PROGRAMAS DE JOGOS DE
GUERRA
Para computadores das famílias TK, Apple, CP e TRS80
2ª EDIÇÃO
EDITORA LUTÉCIA
Programas de jogos
para o seu
microcomputador

PROGRAMAS DE JOGOS DE GUERRA

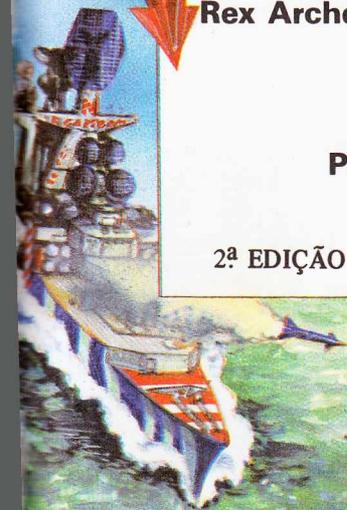
Daniel Isaaman
e Jenny Tyler

Tradução e Adaptação de
Ronaldo Sergio de Biasi

Sumário

Ilustrações
Rex Archer, Jim Bamber, Tony Baskeyfield, Martin Newton, e Graham Round
Diagramação
Graham Round e Roger Priddy
Programa "Invasão dos Robôs": Bob Merry

2ª EDIÇÃO

LUTÉCIA

Introdução

Este livro contém programas de jogos de guerra para microcomputadores. Todos os programas são compatíveis com os micros fabricados no Brasil.

Embora os micros nacionais utilizem a linguagem BASIC, existem três dialetos ou variações dessa linguagem, dependendo da marca e modelo do micro. Os micros nacionais são divididos em três "famílias": a família Sinclair (CP-200, TK-83, TK-85), a família TRS-80 (CP-300, DGT-1000, etc.) e a família Apple (Maxxi, Ap-II, etc.). Se você não souber a que família pertence o seu micro, poderá obter a informação em qualquer loja especializada.

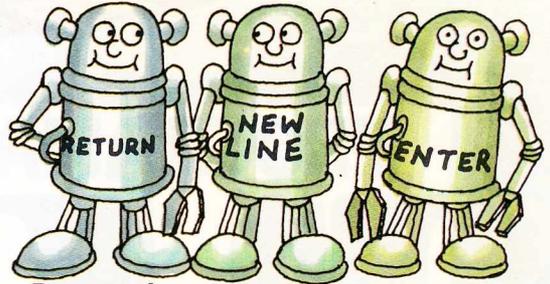
Neste livro, a listagem principal de cada programa é para os micros CP-200 e TK-83 (daqui por diante, tudo o que dissermos do TK-83 também se aplica ao CP-200). As linhas que precisam ser mudadas para que o programa funcione em outros computadores estão indicadas por símbolos e as modificações aparecem no final de cada listagem.

Para cada jogo, há idéias para modificações e extensões dos programas. No final do livro, você encontrará algumas "dicas" para escrever seus próprios programas. Incluímos também uma tabela de equivalência para ajudá-lo a adaptar ao seu micro programas publicados em revistas e em outros livros, além de uma lista das instruções de BASIC usadas neste livro, com as respectivas definições.

Como carregar os programas

Se o seu micro não é o TK-83, as linhas que devem ser mudadas estão indicadas pelos seguintes símbolos:

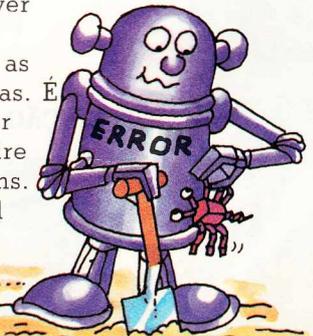
▲TK-85
■família TRS-80
●família Apple

Cada vez que você vir o símbolo da família a que pertence o seu micro, procure no final da listagem uma linha com o mesmo símbolo e o mesmo número. Use-a no lugar da linha correspondente da listagem principal.

Pontos importantes

1 Escreva as linhas exatamente como na listagem, incluindo todos os espaços e sinais de pontuação.
2 Depois de bater cada linha, verifique na tela se está tudo certo.
3 Antes de começar uma nova linha, aperte a tecla RETURN, ENTER ou NEWLINE.
4 Cuidado para não pular uma linha ou misturar duas linhas. Use uma régua ou pedaço de papel para marcar o lugar onde está na listagem.
5 Preste atenção nos símbolos que identificam as diferentes famílias e use a instrução apropriada para o seu modelo de micro.
6 Se você está usando um micro TK-83 ou TK-85, não deve escrever as instruções letra por letra. Em vez disso, use as teclas correspondentes a cada instrução.

Talvez você prefira que alguém leia o programa para você enquanto aperta as teclas. Nesse caso, explique para o seu amigo que ele deve ler todas as vírgulas, pontos, parênteses e espaços.

Como verificar os programas

Depois de carregar o programa, consulte o manual do micro para ver como fazê-lo aparecer na tela. (Em geral, basta escrever LIST.)
Verifique se todas as linhas estão corretas. É muito fácil cometer erros; não se admire se encontrar alguns. Consulte o manual

para saber como alterar as letras ou símbolos que estão errados. Em muitos casos, é mais fácil tornar a escrever a linha inteira. A linha antiga (com o mesmo número) é automaticamente apagada.

Eis uma lista dos erros mais comuns:
1 Linha faltando.
2 Linha com número errado.
3 O começo de uma linha emendado no final da anterior.
4 Falta de parênteses, vírgula, dois pontos, ponto-e-vírgula, ponto ou espaço, especialmente nas linhas mais

compridas. Cuidado com os parênteses duplos.
5 Linha errada para o seu modelo de micro.
6 Troca de zero pela letra "O".
7 Números errados (com zeros a mais, por exemplo).

Experiências com os jogos

Este livro contém várias sugestões para modificar e ampliar os programas, mas não tenha medo de introduzir suas próprias alterações. Não há perigo de danificar o computador; se sua idéia não der certo, você pode sempre voltar ao programa inicial.

Você provavelmente terá vontade de mudar a velocidade de alguns jogos*, especialmente depois que estiver familiarizado com eles. As instruções para isso estão na página de cada programa.

Como executar os programas

Para começar qualquer jogo, você deve escrever RUN. Em alguns jogos, as coisas acontecem muito depressa, de modo que é melhor ler atentamente as instruções antes de começar.

É bastante provável que o programa ainda contenha alguns erros. Nesse caso, não acontecerá nada ou o jogo dará resultados imprevistos. Pode ser que o computador mostre na tela um código de erro, que você pode procurar no manual. Isso o ajudará a localizar o erro. Liste o programa de novo e compare as instruções com as do livro, uma a uma.

Quando o jogo terminar, aparecerá na tela uma mensagem (BREAK IN LINE 200, por exemplo). Para jogar de novo, você deve escrever RUN.

Você pode mudar à vontade o texto entre aspas que aparece depois de qualquer instrução PRINT. Também pode acrescentar novas mensagens. Para isso, escreva o número de uma linha que não aparece no programa (105, por exemplo, se vocc quer incluir uma mensagem

entre as linhas 100 e 110), escreva PRINT e depois o texto da sua mensagem entre aspas.

Se o seu computador é capaz de gerar cores ou sons, consulte o manual para descobrir como funcionam as instruções correspondentes e experimente usá-las nos programas deste livro.

*Na página 35 há uma nota especial para os usuários do TK-85.

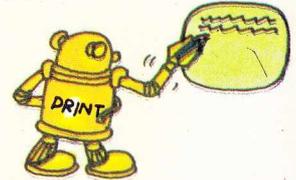
Bomba-relógio

Em 2585, os humanos estão em guerra contra uma raça de robôs. Um míssil lançado pelos robôs acabou de cair e você precisa descobrir o código secreto que desarma a ogiva nuclear. Se não conseguir, o Quartel-General da Terra será destruído.

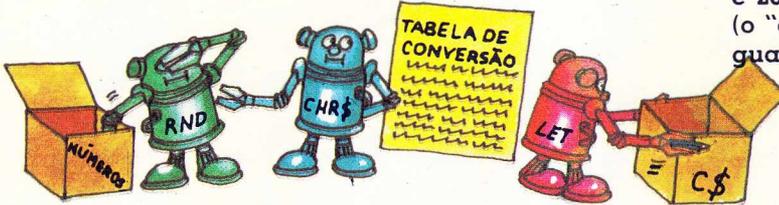
O seu computador sabe qual é a letra correta do código. Entre com um palpite e ele lhe dirá se a letra vem antes ou depois no alfabeto. Você tem direito a quatro tentativas antes de a bomba explodir.

Como o programa funciona

```
10 PRINT "BOMBA-RELOGIO"
20 PRINT
30 PRINT "ESCREVA O CODIGO"
40 PRINT "LETRA (A-Z) PARA"
50 PRINT "DESARMAR A BOMBA."
60 PRINT "VOCE TEM 4 CHANCES"
70 PRINT
```

Imprime o título e as instruções.

```
▲■● 80 LET C$=CHR$(37+INT(RND*26+1))
```

Escolhe um número entre 1 e 26, transforma-o em letra (o "código secreto") e guarda-a em C$.

```
90 FOR G=1 TO 4
```

Começo de um loop que faz com que as linhas de 100 a 140 sejam repetidas 4 vezes.

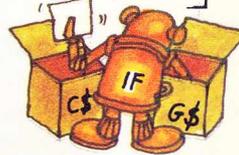
```
100 INPUT G$
```

Guarda seu palpite em G$.

```
110 IF G$=C$ THEN GOTO 210
```

Verifica se você escolheu a letra certa; em caso afirmativo, pula para 210.

```
120 IF G$<C$ THEN PRINT "DEPOIS";
130 IF G$>C$ THEN PRINT "ANTES";
140 PRINT " DE ";G$
```

Verifica se a letra correta fica antes ou depois da que você escolheu e imprime uma mensagem apropriada.

```
150 NEXT G
```

Fim do loop. Volta para outra tentativa.

```
160 PRINT
170 PRINT "BUUUUUUUUMMM..."
180 PRINT "EXPLODIU."
190 PRINT "O CODIGO CERTO ERA ";C$
200 STOP
```

Imprime isto se você errou quatro vezes.

```
210 PRINT "TIC...TIC...TIC..."
220 PRINT "VOCE CONSEGUIU"
230 STOP
```

Imprime isto se você acertou.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
● 80 LET C$=CHR$(64+INT(RND(1)*26+1))
■ 80 LET C$=CHR$(64+INT(RND(0)*26+1))
▲ 80 LET C$=CHR$(64+INT(RND*26+1))
```

Como aumentar o programa

Você pode fazer o computador imprimir uma mensagem especial se você acertar na última tentativa. Acrescente um ponto-e-vírgula no final da linha 220, assim:

220 PRINT "VOCÊ CONSEGUIU";

e substitua a linha 230 por:

230 IF G = 4 THEN PRINT "(EM CIMA DA HORA!)"

Problema

Como você modificaria o programa para aumentar ou diminuir o número de tentativas?

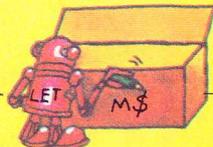
Mensagem Decisiva

Você é um operador de comunicações por laser. Sua missão é interceptar as mensagens dos robôs e transmiti-las para o Quartel-General. Uma mensagem decisiva em código está sendo esperada. Se conseguir transmiti-la corretamente, o ataque dos robôs será rechaçado.

Neste jogo, você precisa guardar na memória um grupo de letras. O computador pede a você um número de dificuldade entre 4 e 10. Quando você entra com a resposta, as letras aparecem na tela e logo desaparecem. Memorize-as e entre com elas no computador, na mesma ordem em que foram mostradas na tela.

Como o programa funciona

```
● 10 CLS
20 PRINT "MENSAGEM DECISIVA"
30 PRINT
40 PRINT "DIFICULDADE? (4-10)"
50 INPUT D
60 IF D<4 OR D>10 THEN GOTO 40
70 LET M$=""
80 FOR I=1 TO D
▲■● 90 LET M$=M$+CHR$(INT(RND*26+38))
100 NEXT I
● 110 CLS
120 PRINT "ENVIE ESTA MENSAGEM:"
130 PRINT
140 PRINT M$
■● 150 FOR I=1 TO D*8
160 NEXT I
● 170 CLS
180 INPUT N$
190 IF N$=M$ THEN GOTO 240
200 PRINT "VOCE ERROU"
210 PRINT "A MENSAGEM ERA:"
220 PRINT M$
230 GOTO 260
240 PRINT "MENSAGEM CORRETA"
250 PRINT "A GUERRA TERMINOU"
260 STOP
```

- 10: Limpa a tela.
- 40–50: Pede um "número de dificuldade" e guarda em D.
- 60: Verifica se o número não é menor que 4 ou maior que 10.
- 70: Define uma string vazia, na qual será guardada a mensagem secreta.
- 80–100: O computador executa este loop D vezes. De cada vez, escolhe uma letra e a acrescenta às que já estão guardadas em **M$**.
- 110–140: Limpa a tela e depois imprime a mensagem.
- 150–160: A mensagem fica na tela enquanto o computador espera durante um tempo que depende de D.
- 170: Limpa a tela no final do loop de espera.
- 180: Guarda sua versão da mensagem em **N$**.
- 190: Verifica se sua mensagem é igual à que está em **M$**; em caso afirmativo, pula para 240.
- 200–220: Chega aqui se você errou e imprime a mensagem correta.

Repare que o TK-83 usa um código diferente para as letras. Todos os outros micros usam o código ASCII.

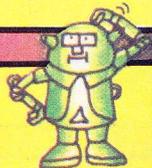
A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
● 10,110,170 HOME
● 90 LET M$=M$+CHR$(INT(RND(1)*26+65))
■ 90 LET M$=M$+CHR$(INT(RND(0)*26+65))
▲ 90 LET M$=M$+CHR$(INT(RND*26+65))
■● 150 FOR I=1 TO D*180
```

Como tornar o jogo mais difícil

Você pode mudar o programa para fazer com que a mensagem secreta contenha também números e sinais de pontuaçao. Para isso, substitua a linha 90 por:

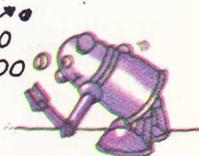
```
90 LET M$ = M$ + CHR$(INT(RND*43 + 21))
● 90 LET M$ = M$ + CHR$(INT(RND(1)*43 + 48))
■ 90 LET M$ = M$ + CHR$(INT(RND(0)*43 + 48))
▲ 90 LET M$ = M$ + CHR$(INT(RND*43 + 48))
```

Problema

Como você modificaria o programa para fazer a mensagem ficar mais tempo na tela?

Duelo

Vocês estão de costas um para o outro. Caminham dez passos, dão meia-volta e sacam as armas. Quem será mais rápido no gatilho?

O computador imprime os números de 1 a 10 para representar os 10 passos, faz uma pausa e imprime ELE SACOU... Você deve apertar uma tecla (qualquer tecla) no momento em que as palavras aparecerem na tela. Se for suficientemente rápido, conseguirá derrotar seu oponente. Não aperte a tecla antes do tempo, caso contrário será automaticamente derrotado.

```
 ●10 CLS
  20 PRINT "DUELO"
  30 PRINT "VOCES ESTAO PRONTOS"
  40 PRINT "VAO DAR 10 PASSOS..."
  50 FOR I=1 TO 10
■●60 PRINT I;"..";
  70 NEXT I
  80 PRINT
■●90 FOR I=1 TO RND*200
 100 NEXT I
●110 IF INKEY$<>"" THEN GOTO 160
 120 PRINT "ELE SACOU..... ";
■●130 FOR I=1 TO 5
●140 IF INKEY$<>"" THEN GOTO 190
 150 NEXT I
```

```
 160 PRINT "ELE ATIROU."
 170 PRINT "VOCE MORREU."
 180 GOTO 210
 190 PRINT "MAS VOCE ATIROU PRIMEIRO"
 200 PRINT "E O MATOU."
 210 STOP
```

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
 ●10 HOME
■●65 FOR J=1 TO 300 : NEXT J
 ●90 FOR I=1 TO RND(1)*1000
 ■90 FOR I=1 TO RND(0)*1000
●110 IF PEEK(-16384)>127 THEN GOTO 160
●130 FOR I=1 TO 20
■130 FOR I=1 TO 50
●140 IF PEEK(-16384)>127 THEN GOTO 190
```

Como mudar a velocidade do jogo

Para mudar o tempo de que você dispõe para apertar uma tecla, basta mudar o último número da linha 130. Quanto menor o número, menor o tempo.

Como tornar o jogo mais difícil

Para incluir a possibilidade de você errar o tiro, faça as seguintes mudanças: Na linha 140, mude 190 para 220. Acrescente as seguintes linhas:

```
  220 IF RND >.3 THEN GOTO 190
● 220 IF RND(1) >.3 THEN GOTO 190
■ 220 IF RND(0) >.3 THEN GOTO 190
  230 PRINT "E VOCÊ ERROU"
  240 GOTO 90
```

Como o programa funciona

— Este loop é repetido 10 vezes. O computador imprime um número e dois pontos de cada vez.

— Este é um loop de espera. O computador passa um certo tempo, que depende do valor de RND, sem fazer nada.

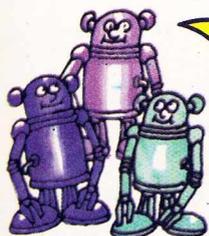
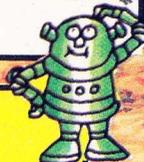
— Verifica se você não se precipitou, apertando uma tecla antes de ELE SACOU aparecer na tela.

— Imprime o sinal de que você pode apertar uma tecla.

— Verifica o teclado para ver se você está apertando uma tecla; em caso afirmativo, pula para 190. (Observe que a linha 140 está no meio de um loop. Isto faz com que o computador verifique o teclado várias vezes, o que lhe dá uma oportunidade razoável de apertar uma tecla.)

— Chega aqui se você perdeu. (Ou se se precipitou ou não apertou nenhuma tecla a tempo.)

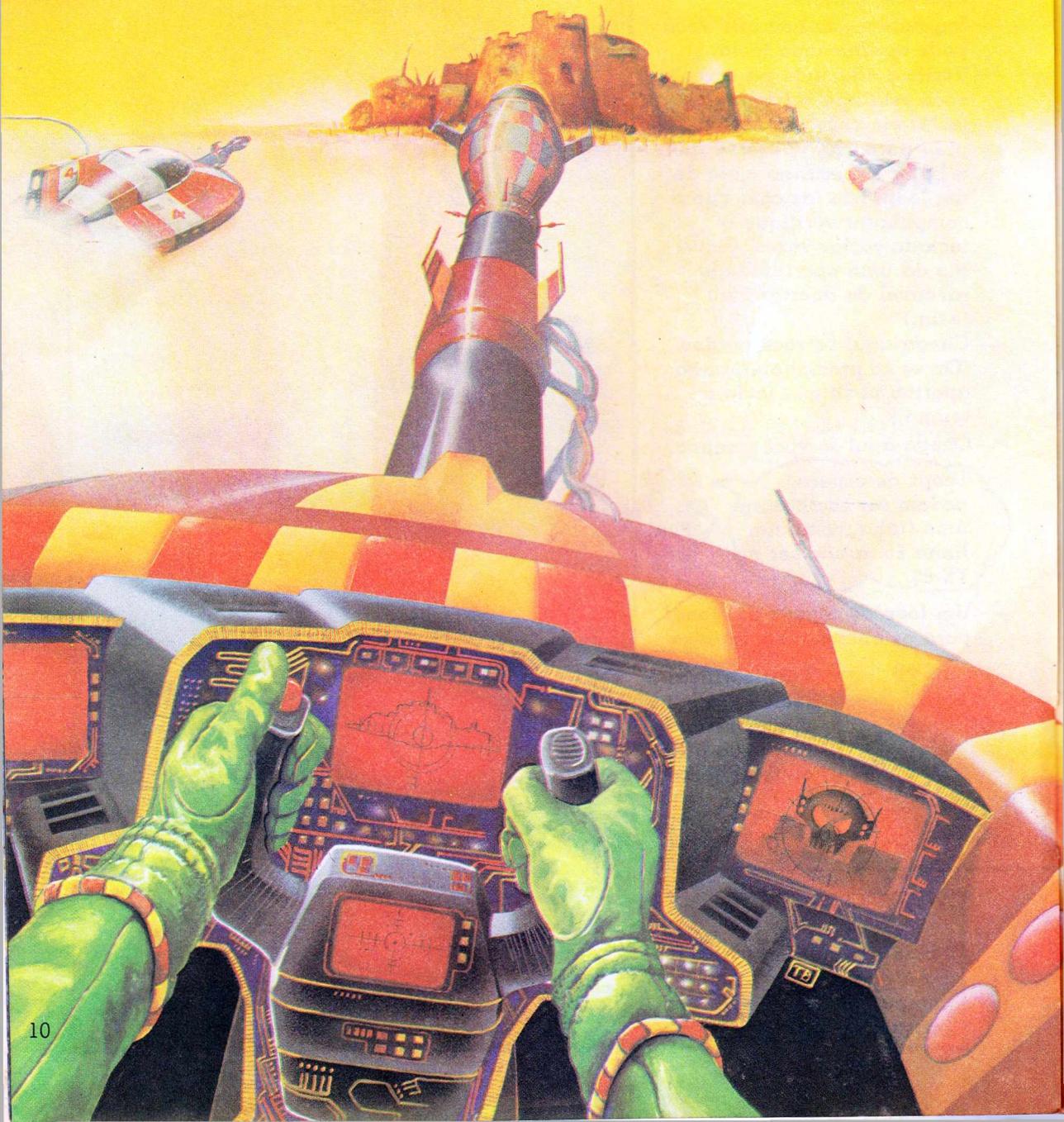
— Chega aqui se você ganhou.

— Um loop de espera adicional para os micros mais rápidos.

Mude o programa para que o computador também possa errar o tiro.

Batalha no Deserto

A última fortaleza dos robôs fora dos EUR* fica nas ruínas de um antigo castelo, no meio do deserto. Você é o comandante do grupo de tanques que foi enviado para destruí-la. Só lhe restam cinco mísseis.

Você precisa calcular a direção e a elevação antes de cada disparo. O computador pede para você fornecer um ângulo de direção entre -90° (tudo para a esquerda) e +90° (tudo para a direita) e um ângulo de elevação entre 0° (horizontal) e 90° (vertical). A elevação determina a que distância do ponto de lançamento o míssil vai cair.

Conseguirá destruir a fortaleza dos robôs?

Como o programa funciona

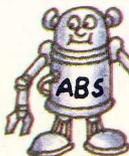
```
   10 PRINT "BATALHA NO DESERTO"
■● 20 LET T=INT(RND*181)-90
■● 30 LET D=RND
   40 FOR G=1 TO 5
   50 PRINT "DIRECAO (-90 A +90) ?"
   60 INPUT T1
   70 PRINT "ELEVACAO (0 A 90) ?"
   80 INPUT B
   90 LET D1=SIN(2*(B/180*3.1416))
   100 IF ABS(T-T1)<2 AND ABS(D-D1)<.05 THEN GOTO 220
   110 PRINT "MISSIL CAIU "
   120 IF T1<T THEN PRINT "A ESQUERDA ";
   130 IF T1>T THEN PRINT "A DIREITA ";
   140 IF ABS(D1-D)>.05 AND T1<>T THEN PRINT "E ";
   150 IF D-D1>.05 THEN PRINT "ANTES";
   160 IF D1-D>.05 THEN PRINT "DEPOIS";
   170 PRINT
   180 NEXT G
   190 PRINT "MISSEIS ACABARAM"
   200 PRINT "ROBOS RESISTIRAM"
   210 STOP
   220 PRINT "*BUUUUUUMMM*"
   230 PRINT "VOCE CONSEGUIU"
   240 STOP
```

Linha 20: Escolhe um número inteiro entre -90 e 90 para a direção.

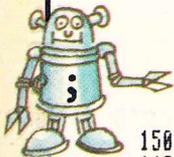
Linha 30: Escolhe um número entre 0 e 1 para a distância do castelo.

Linhas 40–80: Pede seus palpites e os guarda em T1 e B.

Linha 90: Usa o seu ângulo de elevação para calcular a distância percorrida pelo míssil. (A resposta está entre 0 e 1.)

Linha 100: Verifica se você acertou e nesse caso pula para 220.

ABS toma o valor "absoluto" de um número, ignorando o sinal de + ou de - .

Linhas 110–130: Compara o seu ângulo de direção com o número escolhido na linha 20 e imprime uma mensagem apropriada.

Linha 140: Verifica se deve imprimir "E" ou não, comparando a distância que o míssil percorreu com o número escolhido na linha 30.

O ponto-e-vírgula no final de uma instrução PRINT avisa ao computador para *não* mudar de linha.

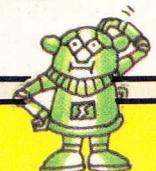
Linhas 150–160: Informa se o míssil caiu antes ou depois do alvo.

Linhas 190–200: Chega aqui se você perdeu.

Linhas 220–230: Chega aqui se você ganhou.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
● 20 LET T=INT(RND(1)*181)-90
■ 20 LET T=INT(RND(0)*181)-90
● 30 LET D=RND(1)
■ 30 LET D=RND(0)
```

Só os micros TK-83 e TK-85 precisam de LET. Nos outros micros, a palavra pode ser omitida.

Problema

Como você faria para incluir a possibilidade de os robôs atirarem em você antes de sua munição acabar?

***Estados Unidos dos Robôs**

O Castelo do Traidor

```
10 PRINT "CASTELO DO TRAIDOR"
20 LET S=0
30 FOR G=1 TO 10
40 LET R$=""
■●50 LET T=INT(RND*9+1)
60 FOR L=1 TO 9
70 IF L=T THEN LET R$=R$+"0"
80 IF L<>T THEN LET R$=R$+"."
90 NEXT L
100 PRINT R$,
■●110 FOR I=1 TO 12
●120 LET I$=INKEY$
130 IF VAL("0"+I$)=T THEN GOTO 170
140 NEXT I
150 PRINT "ERROU"
160 GOTO 190
170 PRINT "NO ALVO"
180 LET S=S+1
190 NEXT G
200 PRINT "VOCE ACERTOU ";S;" EM 10"
210 STOP
```

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
■50 LET T=INT(RND(0)*9+1)
●50 LET T=INT(RND(1)*9+1)
■110 FOR I=1 TO 60
●110 FOR I=1 TO 90
●115 LET I$=""
●120 IF PEEK(-16384)>127 THEN GET I$
```

O Rei está envolvido em uma guerra sangrenta com o seu maior inimigo, o Barão Traidor. Você é um dos arqueiros do Rei e neste exato momento está escondido nos arredores do castelo do Barão, tentando atingir os soldados que se encontram no interior do castelo.

O computador imprime uma fila contendo oito pontos e um 0. As teclas de 1 a 9 correspondem à posição do 0 na fila. Você dispõe de pouco tempo para apertar a tecla certa e atingir o 0 antes que ele desapareça.

Quantos dos soldados do Barão será capaz de derrubar?

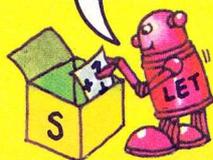
Como mudar a velocidade do jogo

Para mudar o tempo de que você dispõe para apertar uma tecla, basta mudar o último número da linha 110. Quanto menor o número, menor o tempo.

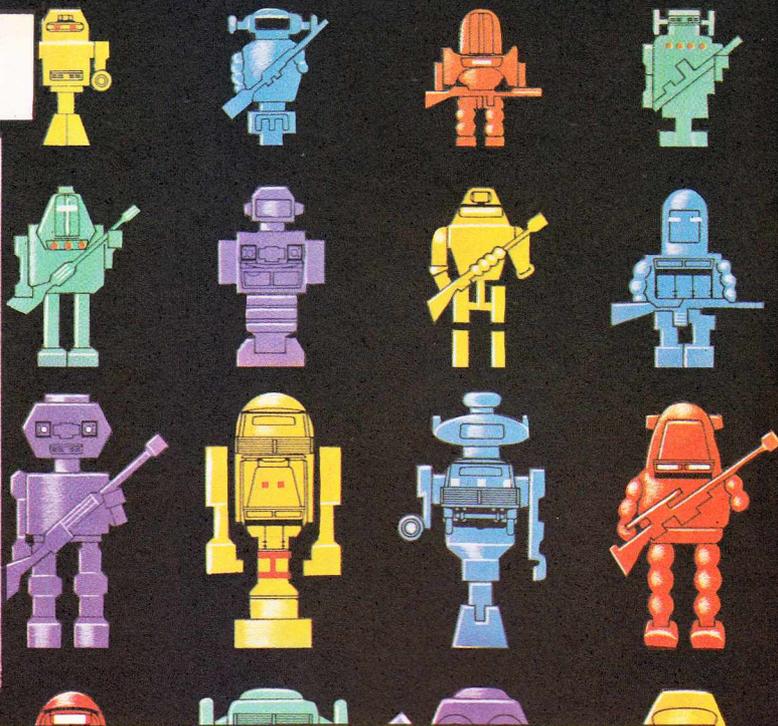
Problema

Mude o programa para que possam aparecer dois tipos de alvos: O (os soldados ordinários do Barão), valendo 1 ponto, e S (os soldados selecionados do Barão), valendo 5 pontos.

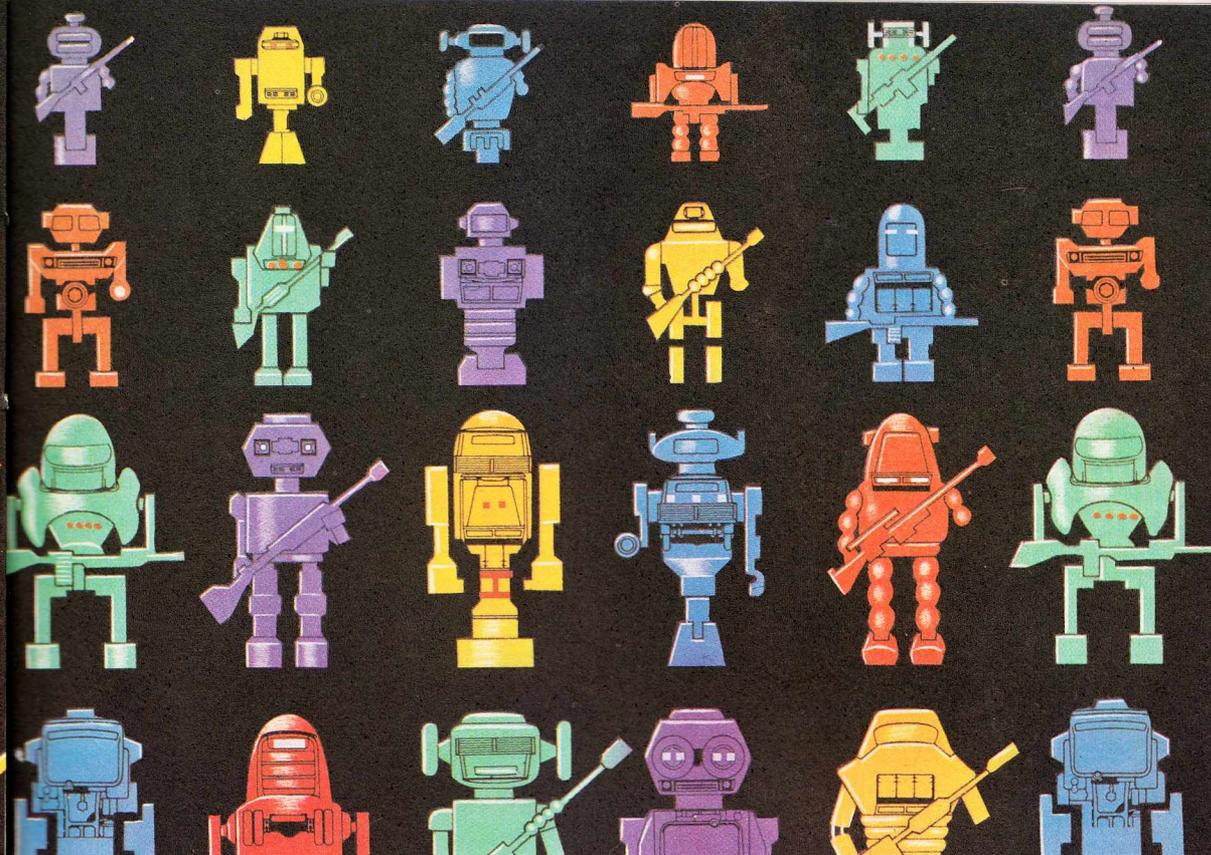
Como o programa funciona

- Inicia a contagem de acertos em zero.
- Começo de um loop de 10 tentativas.
- Cria uma string vazia, R$.
- Escolhe um número entre 1 e 9 e guarda em T.
- Repete este loop 9 vezes, acrescentando de cada vez um caractere à string R$. O caractere é 0 para a posição correspondente ao número que está em T e um ponto para as outras posições.
- Imprime a string R$.
- Espera um certo tempo para ver se você está apertando uma tecla. (Final do loop na linha 140.)
- Verifica se está apertando a tecla certa. Em caso afirmativo, pula para 170.
- Chega aqui se você apertou a tecla errada ou não apertou nenhuma tecla. Imprime a mensagem e pula para 190.
- Aumenta a sua contagem de 1.
- Volta para a linha 30 para nova tentativa.
- Imprime a contagem depois de 10 tentativas.

Invasão dos Robôs

A Terra está sendo invadida por robôs de vários tipos. Você dispõe de muitas armas, mas cada uma só faz efeito contra um certo tipo de robô.

Um símbolo de código, identificando o robô atacante, aparece na tela. Aperte imediatamente a tecla com o mesmo símbolo — em alguns casos é preciso apertar também a tecla SHIFT — e veja quantos robôs consegue destruir.

```
     10 PRINT "INVASAO DOS ROBOS"
     20 LET H=0
     30 FOR T=1 TO 25
  ■● 40 FOR I=1 TO INT(RND *30+20)
     50 NEXT I
  ■● 60 LET A=INT(RND*20)
  ■● 70 LET D=INT(RND*15)
 ▲■● 80 LET P$=CHR$(INT(RND*53+11))
   ● 90 CLS
    100 FOR J=0 TO D
    110 PRINT
    120 NEXT J
    130 PRINT TAB(A);P$
 ■● 140 FOR I=1 TO 15
●→● 150 LET R$=INKEY$
    160 IF R$=P$ THEN GOTO 210
    170 IF R$<>"" THEN GOTO 190
    180 NEXT I
    190 PRINT "ERROU"
    200 GOTO 230
    210 PRINT "ACERTOU"
    220 LET H=H+1
    230 NEXT T
  ● 240 CLS
    250 PRINT "VOCE ACERTOU ";H;" EM 25"
    260 STOP
```

Como o programa funciona

- Linha 20: Inicia a contagem de acertos em zero.
- Linha 30: Começo de um loop de 25 tentativas.
- Linhas 40–50: Espera variável.
- Linhas 60–70: Escolhe uma posição na tela.
- Linha 80: Escolhe um caractere.
- Linha 90: Limpa a tela.
- Linhas 100–120: Faz o cursor descer, linha por linha, até chegar à linha número D (que foi escolhida na linha 70).
- Linha 130: Faz o cursor se deslocar A espaços e imprime o caractere escolhido na linha 80.
- Linhas 140–180: Verifica se você está apertando alguma tecla e se é a tecla certa. Em seguida, salta para a linha apropriada.
- Linha 220: Aumenta a contagem de 1.
- Linha 230: Volta para outra tentativa.
- Linha 250: Imprime a contagem depois de 25 tentativas.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
● 40 FOR I=1 TO INT(RND(1)*300+200)
■ 40 FOR I=1 TO INT(RND(0)*300+200)
● 60,70 mude RND para RND(1)
■ 60,70 mude RND para RND(0)
● 80 LET P$=CHR$(INT(RND(1)*58+33))
■ 80 LET P$=CHR$(INT(RND(0)*58+33))
▲ 80 LET P$=CHR$(INT(RND*58+33))
● 90,240 HOME
■● 140 FOR I=1 TO 150
● 145 R$=""
● 150 IF PEEK(-16384)>127 THEN GET R$
```

Como aumentar a velocidade do jogo

Para tornar o jogo mais rápido, substitua o último número da linha 140 por um número menor.

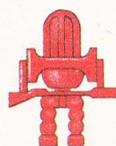
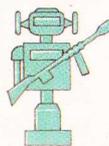
Movimento do cursor

Talvez haja uma maneira mais simples de mover o cursor para um determinado ponto da tela (linhas 100-130). Consulte o manual do seu micro.

Problema

Estes são os robôs mais perigosos.

U
Mecanos

V
Latóides

W
Galipões

X
Sucatinos

Y
Charangos

Modifique o programa para ganhar 100 pontos pelos robôs acima e apenas 10 pontos pelos outros.

Arma Secreta

Se você conseguir destruir o depósito subterrâneo de peças dos robôs, que fica em algum lugar dos EUR, os humanos vencerão a guerra.

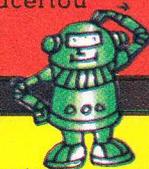
Você dispõe de uma arma secreta que é capaz de abrir buracos no chão, vaporizando tudo o que encontra. Entretanto, o depósito está muito bem escondido. Só lhe resta disparar a arma cegamente, na esperança de atingir o alvo.

O computador pede a você um número de dificuldade (o menor número permitido é 4) e depois lhe pergunta quais são as coordenadas X e Y do alvo. (Entre com as coordenadas separadamente, apertando RETURN, ENTER ou NEWLINE depois de cada número.)

Para saber quais são os valores possíveis de X e Y, leia com atenção a listagem do programa.

```
● 10 CLS
  20 PRINT "ARMA SECRETA"
  30 PRINT "DIGA A DIFICULDADE"
  40 INPUT D
  50 IF D<4 THEN GOTO 30
■● 60 LET X=INT(RND*D+1)
■● 70 LET Y=INT(RND*D+1)
  80 FOR G=1 TO D+5
  90 PRINT "PALPITES PARA X E Y"
 100 INPUT X1
 110 INPUT Y1
 120 LET Z=SQR((X-X1)*(X-X1)+
     (Y-Y1)*(Y-Y1))
 130 IF Z=0 THEN GOTO 200
 140 IF Z<=3 THEN PRINT "PASSOU PERTO"
 150 IF Z>3 THEN PRINT "PASSOU LONGE"
 160 NEXT G
 170 PRINT "OS ROBOS VIRAM VOCE"
 180 PRINT "AHHHHHHHHHHH......"
 190 STOP
 200 PRINT "VOCE VENCEU DEPOIS"
 210 PRINT "DE ";G;" TENTATIVAS"
 220 STOP
```


Como o programa funciona

- (linhas 30–50) Pede um número de dificuldade, guarda em D e verifica se não é menor que 4.
- (linhas 60–70) Escolhe números para X e Y.
- (linha 80) Começo de um loop que lhe dá um número de tentativas que depende da dificuldade escolhida.
- (linhas 90–110) Pede seus palpites e guarda em X1 e Y1.
- (linha 120) Calcula a distância entre o seu tiro e o alvo e guarda a resposta em Z.
- (linhas 130–150) Consulta Z para saber se o tiro passou perto e imprime uma mensagem. (Se Z é zero, você acertou no alvo.)
- (linha 160) Fim do loop. Volta para outra tentativa.
- (linhas 170–180) Chega aqui se você esgotou as tentativas.
- (linhas 200–210) Chega aqui se você acertou no alvo.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
●10 HOME
●60 LET X=INT(RND(1)*D+1)
■60 LET X=INT(RND(0)*D+1)
●70 LET Y=INT(RND(1)*D+1)
■70 LET Y=INT(RND(0)*D+1)
```

Problema

Veja se consegue acrescentar o seguinte sistema de contagem:

1 ponto cada vez que o tiro passar perto.

10 pontos cada se você acertar no alvo.

Fuga!

Os robôs o capturaram e o trancaram em uma cela, depois de lhe tomar todas as armas. De repente, você se lembra de que ainda está com o relógio de sonar, que pode ser ajustado para gerar qualquer freqüência. Se conseguir encontrar a freqüência de ressonância dos seus guardas robôs, poderá destruí-los.

Você precisa tomar cuidado para não usar freqüências muito baixas, caso contrário o edifício poderá desabar, soterrando-o. Por outro lado, se a freqüência for alta demais, sua cabeça doerá tanto que você terá que desistir.

Conseguirá escapar aos horrores da prisão dos robôs? (Para saber quais são os valores de freqüência que deve escolher, leia com atenção a listagem do programa.

```
●10 CLS
 20 PRINT "FUGA"
■●30 LET F=INT(RND*100+1)
 40 LET L=1
 50 LET H=1
 60 FOR G=1 TO 5
 70 PRINT "PALPITE ";
 80 INPUT F1
 90 IF ABS(F-F1)<5 THEN GOTO 290
100 IF F-F1>40 THEN GOTO 170
110 IF F1-F>40 THEN GOTO 230
120 PRINT "NENHUM EFEITO"
130 NEXT G
140 PRINT "VOCE DEMOROU DEMAIS"
150 PRINT "A FREQUENCIA ERA ";F
160 STOP
170 IF L=2 THEN GOTO 210
180 PRINT "GRAVE DEMAIS...CUIDADO"
190 LET L=2
200 GOTO 130
210 PRINT "O EDIFICIO DESABOU"
220 STOP
230 IF H=2 THEN GOTO 270
240 PRINT "AGUDO DEMAIS...AI"
250 LET H=2
260 GOTO 130
270 PRINT "SUA CABECA DOI...DESISTA"
280 STOP
290 PRINT "VOCE CONSEGUIU"
300 STOP
```

Como o programa funciona

Linha 30: Escolhe um número entre 1 e 100 para a freqüência dos robôs e guarda em F.

Linhas 40–50: Guarda 1 em L e H. Essas variáveis são usadas quando você escolhe uma freqüência muito alta ou muito baixa – veja linhas 170 a 190 e 230 a 250.

Linha 60: Começo de um loop de 5 tentativas.

Linhas 70–80: Pede a você um palpite e guarda em F1.

Linha 90: Verifica se seu palpite está muito próximo de F. Se está, pula para 290 para imprimir VOCÊ CONSEGUIU.

Linha 100: Pula para 170 se o seu palpite está mais de 40 abaixo de F.

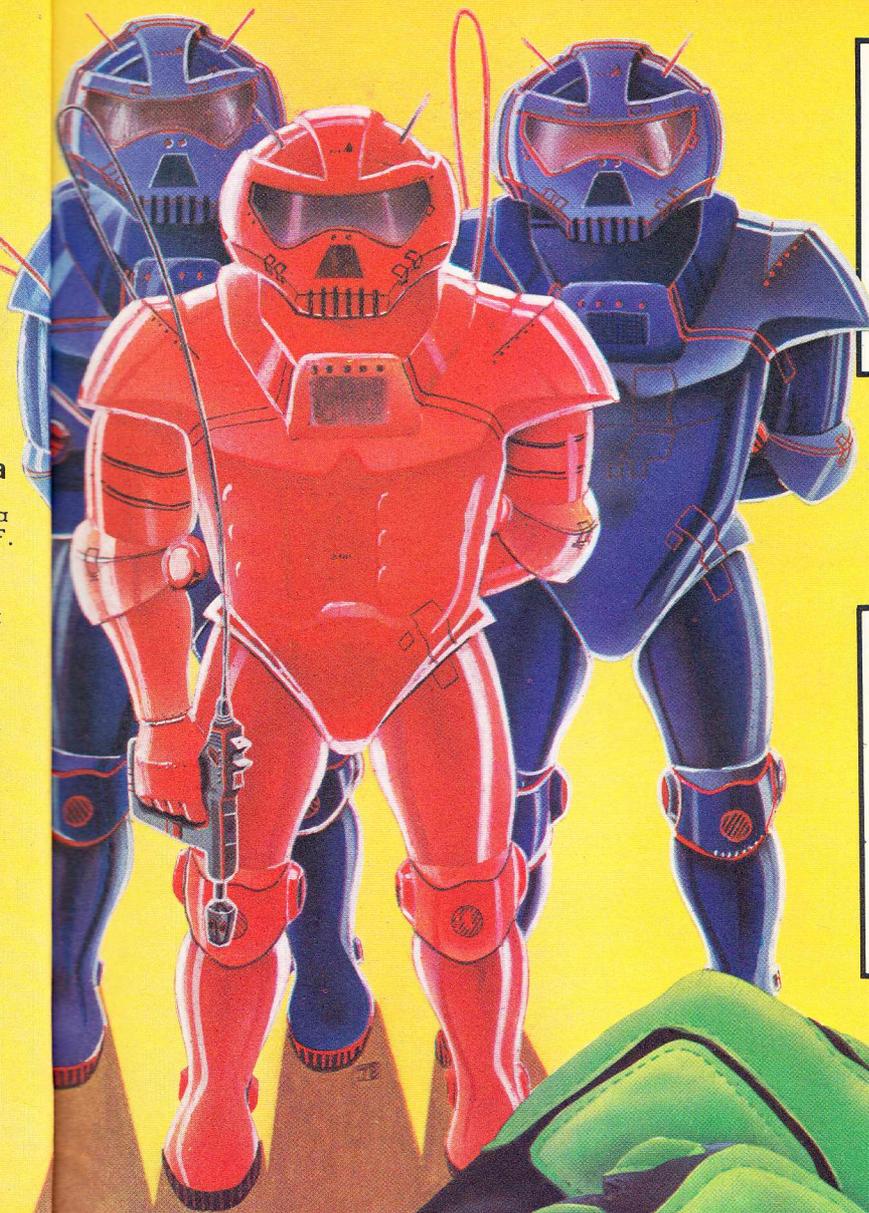
Linha 110: Pula para 230 se o seu palpite está mais de 40 acima de F.

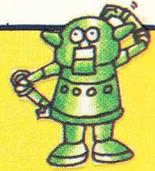
Linhas 120–160: Chega aqui se o seu palpite está razoavelmente próximo de F e volta para outra tentativa. Se as tentativas acabaram, imprime a freqüência correta.

Linhas 170–220: Verifica o valor de L. Na primeira vez que o programa chega aqui, L é 1. O computador imprime uma advertência, faz L igual a 2 e volta para outra tentativa. Na vez seguinte, o programa pula direto para 210 e informa que você perdeu.

Linhas 230–280: Verifica o valor de H. Na primeira vez, adverte que a freqüência está alta demais; na segunda, informa que você perdeu.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
●10 HOME
●30 LET F=INT(RND(1)*100+1)
■30 LET F=INT(RND(0)*100+1)
```

Problema

Cada um dos três guardas robôs tem uma freqüência de ressonância diferente. Você só pode escapar depois de destruir os três. Mude o programa para que ele funcione desta forma.

Como tornar o jogo mais difícil

Mude o 5 na linha 90 para um número menor. Isto significa que você terá que chegar mais perto de F para ganhar. Você também pode aumentar a faixa de valores possíveis de F mudando o 100 da linha 30 para um número maior.

Avião Pirata

É você contra o Pirata Negro. Ele se adianta, você acelera; ele fica para trás, você freia. Você deve emparelhar com ele e atirar, torcendo para que ele não dispare primeiro.

Use as teclas A para acelerar, D para frear e F para atirar. O computador lhe dirá qual a sua posição e velocidade em relação ao pirata. Cuidado para não deixar o pirata se afastar demais de você.

Como o programa funciona

```
10 CLS
20 PRINT "AVIAO PIRATA"
30 LET V=INT(RND*11-5)
```

Escolhe um número entre -5 e +5 para sua velocidade em relação ao pirata e guarda em V.

```
40 LET S=-INT(RND*3+1)
```

Escolhe um número para sua distância do pirata e guarda em S. No início o número é negativo e portanto você está atrás dele.

```
50 IF ABS(S)>20 THEN GOTO 230
```

Verifica se a distância entre vocês é maior que 20. Em caso afirmativo, pula para 230 para informar que você o perdeu de vista.

```
60 CLS
70 PRINT "VOCE ESTA ";
80 IF S<0 THEN PRINT "ATRAS"
90 IF S>0 THEN PRINT "A FRENTE"
100 IF S=0 THEN PRINT "JUNTO"
110 PRINT "SUA VELOCIDADE E ";
120 IF V>0 THEN PRINT "MAIOR"
130 IF V<0 THEN PRINT "MENOR"
140 IF V=0 THEN PRINT "IGUAL"
```

Verifica os valores de S e V e imprime sua posição e velocidade em relação ao pirata.

```
150 LET I$=INKEY$
160 IF I$="A" THEN LET V=V+1
170 IF I$="D" THEN LET V=V-1
180 IF I$="F" AND S=0 THEN GOTO 250
```

Verifica qual a tecla que você está apertando. Se é A, aumenta sua velocidade de 1. Se é D, diminui sua velocidade de 1. Se é F e se S=0 (isto é, se você está emparelhado com o pirata), pula para 250. (Se S não é zero e você aperta F, nada acontece.)

```
190 LET S=S+V
```

Calcula a nova distância entre vocês.

```
■●200 FOR I=1 TO 20
  210 NEXT I
  220 GOTO 50
  230 PRINT "VOCE O PERDEU DE VISTA"
  240 GOTO 330
  250 IF ABS(V)<2 THEN GOTO 290
  260 PRINT "VOCE SO CONSEGUIU"
  270 PRINT "ASSUSTA-LO"
  280 GOTO 40
■●290 IF RND>.7 THEN GOTO 320
  300 PRINT "VOCE O ABATEU"
  310 GOTO 330
  320 PRINT "ELE ATIROU PRIMEIRO"
  330 STOP
```

- Linhas 200–210: Loop de espera.
- Linha 220: Volta a 50 para repetir.
- Linha 250: Chega aqui se você atirou e estava emparelhado com o pirata. Verifica se a velocidade é menor que 2 (ignorando o sinal). Em caso afirmativo, pula para 290.
- Linhas 260–280: Imprime uma mensagem se a velocidade era maior que 2 quando você disparou. Volta a 40 para recomeçar com um novo valor de S.
- Linha 290: Sorteia para ver se você atirou antes do pirata.
- Linhas 300–330: Imprime uma mensagem adequada, de acordo com o resultado do sorteio da linha 290.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
●10,60 HOME
●30 LET V=INT(RND(1)*11-5)
■30 LET V=INT(RND(0)*11-5)
●40 LET S=-INT(RND(1)*3+1)
■40 LET S=-INT(RND(0)*3+1)
●145 I$=""
●150 IF PEEK(-16384)>127 THEN GET I$
■●200 FOR I=1 TO 200
■290 IF RND(0)>.7 THEN GOTO 320
●290 IF RND(1)>.7 THEN GOTO 320
```

Como tornar o jogo mais fácil

Este jogo é bastante difícil. Para torná-lo um pouco mais fácil, acrescente as linhas abaixo. Assim você poderá ver na tela a posição relativa dos dois aviões.

```
195 IF ABS(S)>10 THEN GOTO 200
196 PRINT TAB(w/2);"ELE"
197 PRINT TAB(S+w/2);"VOCE"
```

Substitua w pela largura da tela.

Problema

O sorteio da linha 290 é favorável a você. Sabe como mudá-lo para que você e o pirata tenham a mesma probabilidade de vencer?

Bombardeiro Supersônico

Você está sobrevoando os EUR em uma missão de bombardeio. O computador de bordo mostra um gráfico da população de robôs nas várias cidades, baseado em fotografias tiradas por satélites. Você só tem tempo de atacar um alvo em cada cinco, de modo que deve escolher rapidamente a cidade que tem a maior população de robôs e lançar sobre ela uma das suas bombas de "Corrodapalm". (Essas bombas contêm uma substância tão corrosiva que pode dissolver o corpo de um robô em questão de segundos.)

Para lançar uma bomba, aperte a tecla com o número que está ao lado da cidade com maior população de robôs. No caso de duas populações iguais, escolha o número menor.

Conseguirá voltar à base como herói?

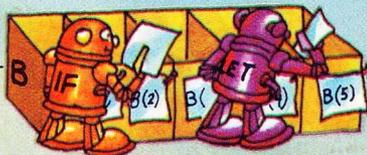
Como o programa funciona

```
●10 CLS
 20 PRINT "BOMBARDEIRO SUPERSONICO"
 30 DIM B(5)
```

Esta instrucão define B como uma variável indexada, que pode guardar cinco números diferentes, identificados como B(1), B(2), B(3), B(4) e B(5).

```
 40 LET S=0
```

Inicia a contagem dos acertos em zerc.

```
 50 FOR G=10 TO
    1 STEP -1
```

Começa um loop de 10 tentativas. O índice do loop varia ao contrario do normal (G = 10 da primeira vez, 9 da segunda vez, etc.) Com isso, o loop de espera da linha 190 reserva mais tempo para as primeiras tentativas.

```
 60 LET M=1
```

Coloca a maior populaçáo em 1 para começar.

```
   70 FOR I=1 TO 5
■● 80 LET B(I)=INT(RND*10+1)
   90 IF B(I)>B(M) THEN LET M=I
  100 NEXT I
```

Escolhe 5 números e guarda em B(1) a B(5). Verifica qual é o maior e muda M para o valor apropriado.

```
 ●110 CLS
  120 FOR I=1 TO 5
  130 PRINT I;
  140 FOR J=1 TO B(I)
  150 PRINT "**";
  160 NEXT J
  170 PRINT
  180 NEXT I
```

Imprime os 5 números na tela na forma de filas de asteriscos.

```
■●190 FOR I=1 TO G*3
●→
 ●200 LET I$=INKEY$
  210 IF I$<>"" THEN GOTO 270
  220 NEXT I
```

Verifica se você está apertando uma tecla; em caso afirmativo, vai para 270.

```
  230 PRINT "TARDE DEMAIS"
■●240 FOR J=1 TO 10
  250 NEXT J
  260 GOTO 280
```

Se você não apertar nenhuma tecla a tempo, imprime TARDE DEMAIS e volta para outra tentativa.

```
  270 IF VAL(I$)=M THEN LET S=S+1
```

Verifica se você apertou a tecla certa; em caso afirmativo, aumenta a contagem de 1.

```
  280 NEXT G
```

Volta para outra tentativa.

```
  290 PRINT "VOCE ACERTOU ";S;" EM 10"
  300 PRINT "ALVOS PREFERENCIAIS"
  310 IF S=10 THEN PRINT "VOCE E UM HEROI"
  320 IF S<10 THEN PRINT "PENA...VOCE FRACASSOU"
  330 STOP
```

Imprime a contagem depois de 10 tentativas.

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
 ● 10,110 HOME
 ● 80 LET B(I)=INT(RND(1)*10+1)
 ■ 80 LET B(I)=INT(RND(0)*10+1)
■● 190 FOR I=1 TO G*30
 ● 195 I$=""
 ● 200 IF PEEK(-16384)>127 THEN GET
■● 240 FOR J=1 TO 400
```

Como mudar a velocidade do jogo

Para ter mais tempo para apertar uma tecla, troque o último número da linha 190 para um maior.

Quando tiver mais prática, diminua o número da linha 190. Verifique até onde consegue chegar.

Problema

Você sabe como mudar o programa para aumentar o número de alvos?

"Iceberg"

Seu navio está muito avariado e a munição acabou. Quando está voltando lentamente para casa, navegando em águas traiçoeiras, coalhadas de *icebergs*, você descobre que está sendo seguido por um navio inimigo. Curiosamente, ele pode detectá-lo, mas não aos *icebergs*; assim, você deve fazer tudo para que se choque com um deles.

O computador mostra a posição do seu navio (Y), do inimigo (Z) e dos *icebergs* (*). Você pode andar um espaço para o norte, sul, leste ou oeste de cada vez. O inimigo sempre se dirige para você pelo caminho mais curto (ele também pode se mover em diagonal). Se você ficar em uma das 8 posições em volta do inimigo, será capturado; se bater em um *iceberg*, afundará.

Conseguirá escapar e voltar para casa?

Como o programa funciona

```
   10 PRINT "ICEBERG"
   20 DIM B(8,8)
■● 30 LET N=INT(RND*8+4)
   40 FOR I=1 TO N
▲■● 50 LET B(INT(RND*8+1),INT(RND*8+1))=23
   60 NEXT I
■●  70 LET SX=INT(RND*8+1)
■●  80 LET SY=INT(RND*8+1)
    90 IF B(SX,SY)<>0 THEN GOTO 70
▲■● 100 LET B(SX,SY)=63
■● 110 LET YX=INT(RND*8+1)
■● 120 LET YY=INT(RND*8+1)
   130 IF B(YX,YY)<>0 THEN GOTO 110
▲■● 140 LET B(YX,YY)=62
●  150 CLS
   160 FOR Y=1 TO 8
   170 FOR X=1 TO 8
   180 IF B(X,Y)=0 THEN GOTO 210
   190 PRINT CHR$(B(X,Y));
   200 GOTO 220
   210 PRINT ".";
   220 PRINT " ";
   230 NEXT X
   240 PRINT
   250 NEXT Y
```

B é uma variável indexada com dimensões 8 por 8. (linha 20)

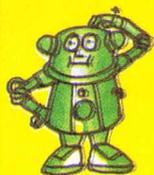
N é o número de *icebergs*, que varia de 4 a 11, dependendo do valor de RND. (linha 30)

23 é o código para o asterisco no TK-83. O computador repete o loop N vezes, colocando asteriscos em pontos aleatórios do mapa. (linhas 40–60)

Coloca o navio do inimigo no mapa para começar o jogo. 63 é o código para a letra Z no TK-83. A linha 90 verifica se existe um *iceberg* na posição escolhida; em caso afirmativo, escolhe outra posição. (linhas 70–100)

Coloca o seu navio no mapa. 62 é o código para a letra Y no TK-83. (linhas 110–140)

Imprime o mapa com as posições correntes dos navios e *icebergs*. (linhas 150–240)

Problema

Você sabe como aumentar o tamanho do mapa?
Veja se consegue também aumentar o número de *icebergs*.

```
260 LET B(YX,YY)=0
```
Apaga sua posição corrente do mapa para que você possa especificar uma nova posição.

```
270 PRINT "DIRECAO (N,S,L,O) "
280 INPUT D$
```
Guarda sua direção em DS.

```
■ 290 LET YY=YY+(D$="S" AND YY<>8)
■ 300 LET YY=YY-(D$="N" AND YY<>1)
■ 310 LET YX=YX+(D$="L" AND YX<>8)
■ 320 LET YX=YX-(D$="O" AND YX<>1)
```
Calcula sua nova posição, tomando cuidado para que não caia fora do mapa.

```
▲■● 330 IF B(YX,YY)=63 THEN GOTO 500
▲■● 340 IF B(YX,YY)=23 THEN GOTO 600
```
Verifica sua nova posição para ver se você colidiu com o navio inimigo ou com um *iceberg*. Em caso afirmativo, pula para imprimir uma mensagem apropriada.

```
▲■● 350 LET B(YX,YY)=62
    360 LET B(SX,SY)=0
```
Coloca o símbolo do seu navio na nova posição.

```
    370 LET SX=SX+SGN(YX-SX)
    380 LET SY=SY+SGN(YY-SY)
```
Calcula a nova posição do navio inimigo.

```
▲■● 390 IF B(SX,SY)=62 THEN GOTO 500
▲■● 400 IF B(SX,SY)=23 THEN GOTO 700
```
Verifica a nova posição do navio inimigo para ver se capturou você ou colidiu com um *iceberg*. Em caso afirmativo, pula para imprimir uma mensagem apropriada.

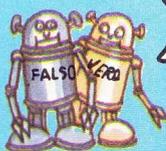
```
▲■● 410 LET B(SX,SY)=63
```
Coloca o símbolo do navio inimigo na nova posição.

```
    420 GOTO 150
```
Volta a 150 para começar uma nova jogada.

```
    500 PRINT "VOCE FOI APANHADO"
    510 GOTO 800
    600 PRINT "VOCE BATEU NUM ICEBERG"
    610 GOTO 800
    700 PRINT "VOCE ESTA SALVO"
    710 PRINT "ELE BATEU NUM ICEBERG"
    800 STOP
```

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
●   30,50,70,80,110,120 mude RND para RND(1)
■   30,50,70,80,110,120 mude RND para RND(0)
▲■● 50,340,400 mude 23 para 42
▲■● 100,330,410 mude 63 para 90
▲■● 140,350,390 mude 62 para 89
●   150 HOME
■   290 LET YY=YY-(D$="S" AND YY<>8)
■   300 LET YY=YY+(D$="N" AND YY<>1)
■   310 LET YX=YX-(D$="L" AND YX<>8)
■   320 LET YX=YX+(D$="O" AND YX<>1)
```

Os computadores podem **verificar se uma expressão é verdadeira ou falsa.** Se é falsa, eles dão o valor 0. Se é verdadeira, alguns computadores dão o valor +1, outros -1. Por isso, as linhas 290-320 têm que ser mudadas em alguns micros.

O Muro do Bruxo

O muro que o Bruxo construiu em volta de seu castelo não é um muro comum. É feito de pessoas, paralisadas pelo olhar do Bruxo, e além disso pode se mover para a frente e para trás. Se você conseguir abrir um buraco nele com o auxílio de sua fiel catapulta, capaz de arremessar pedras gigantescas, anulará os poderes mágicos do Bruxo e devolverá à vida os prisioneiros. Entretanto, isto não é tão fácil como parece; o Bruxo está trabalhando em um feitiço capaz de rebater os tiros de volta e às vezes é capaz de fazê-lo funcionar...

Se você fracassar, também passará a fazer parte do muro do Bruxo!

```
 10 CLS
 20 PRINT "MURO DO BRUXO"
 30 PRINT
 40 PRINT "QUER INSTRUCOES?"
 50 INPUT I$
 60 IF I$(1)="S" THEN GOSUB 740
 70 PRINT "DIFICULDADE?"
 80 PRINT "(5=FACIL, 1=DIFICIL)"
 90 INPUT Q
100 DIM W(8,8)
110 DIM E(8)
120 FOR Y=1 TO 8
130 FOR X=1 TO 4
140 LET W(X,Y)=20
150 NEXT X
160 NEXT Y
170 LET Z=0
180 FOR C=1 TO 3
190 LET D=INT(RND*80+21)
200 GOSUB 430
210 IF Z=1 THEN GOTO 1150
220 PRINT "VOCE ESTA A ";D;" METROS DE DISTANCIA"
230 LET W=INT(RND*41)-20
240 IF W=0 THEN PRINT "SEM VENTO"
250 IF W<0 THEN PRINT "VENTO PARA A DIREITA"
260 IF W>0 THEN PRINT "VENTO PARA A ESQUERDA"
270 PRINT "DIGA A ELEVACAO (1-90)"
280 INPUT A
290 PRINT "DIGA A VELOCIDADE"
```

```
     300 INPUT V
     310 LET A=A/180*3.1416
     320 LET H=TAN(A)*(D-W)-(5*(D-W)**2)/(V*COS(A))**2
     330 LET H=INT(H/Q)
     340 IF H>0 AND H<9 THEN GOSUB 540
     350 IF H<1 THEN PRINT "TIRO CAIU ANTES"
     360 IF H<8 THEN PRINT "TIRO CAIU DEPOIS"
■●   370 IF RND>.2 THEN GOTO 200
     380 PRINT "O MURO SE MEXEU..."
     390 NEXT C
     400 PRINT "VOCE FOI TRANSFORMADO"
     410 PRINT "EM PEDRA"
     420 STOP
■●   430 FOR I=1 TO 30
     440 NEXT I
 ●   450 CLS
     460 PRINT
     470 FOR Y=8 TO 1 STEP -1
     480 FOR X=1 TO 8
■●   490 PRINT CHR$(W(X,Y));
     500 NEXT X
     510 PRINT
     520 NEXT Y
     530 RETURN
     540 GOSUB 650
     550 LET W(E(H)-1,H)=0
     560 IF V*COS(A)>50 THEN LET W(E(H)-2,H)=0
■●   570 IF RND>.5 AND H<>1 THEN LET W(E(1),1)=29
■●   580 IF RND>.5 AND H>5 AND H<8 THEN LET
         W(E(H+1)-1,H+1)=0
     590 IF H>1 THEN GOTO 630
     600 FOR Y=2 TO 8
■●   610 IF RND<.5 THEN LET W(E(Y)-1,Y)=0
     620 NEXT Y
     630 GOSUB 650
     640 RETURN
     650 FOR Y=1 TO 8
     660 LET X=1
     670 IF W(X,Y)=0 OR X=8 THEN GOTO 700
     680 LET X=X+1
     690 GOTO 670
     700 LET E(Y)=X
     710 IF X=1 THEN LET Z=1
```

SUBSTITUA ** NA LINHA 320
POR ^ PARA O APPLE E
POR ↑ PARA O TRS-80 E
PARA O TK-85

O programa continua na proxima pag...

O Muro do Bruxo (continuação)

```
720 NEXT Y
730 RETURN
740 PRINT "VOCE ESTA ATACANDO O"
750 PRINT "ULTIMO REDUTO DE UM"
760 PRINT "BRUXO FAMOSO, QUE"
770 PRINT "ESTA ESCONDIDO ATRAS"
780 PRINT "DE UM MURO DE PEDRA."
790 PRINT "CADA PEDRA E UMA"
800 PRINT "VITIMA DO BRUXO."
810 PRINT "SO VOCE PODE ATACAR"
820 PRINT "E LIBERTA-LOS"
830 PRINT "DA MAGIA DO BRUXO."
840 PRINT "VOCE DEVE DESTRUIR A"
850 PRINT "PAREDE USANDO CATAPULTAS,"
860 PRINT "MAS CUIDADO, O BRUXO"
870 PRINT "PODE MOVER O MURO"
880 PRINT "PARA A FRENTE E PARA"
890 GOSUB 1110
900 PRINT "TRAS, E UMA VEZ OU OUTRA"
910 PRINT "REBATER SEUS TIROS"
920 PRINT "DE VOLTA."
930 PRINT "DEPOIS DE CADA TIRO, VOCE"
940 PRINT "VE UMA PLANTA DO MURO"
950 PRINT "MOSTRANDO OS DANOS"
960 PRINT "QUE JA CONSEGUIU"
970 PRINT "CAUSAR."
980 PRINT "REPARE QUE HA CERTAS"
990 PRINT "PEDRAS QUE PRODUZEM"
1000 PRINT "MAIS DANOS, E TAMBEM QUE,"
1010 PRINT "QUANTO MAIS DEPRESSA O"
1020 PRINT "PROJETIL ESTIVER SE"
1030 PRINT "MOVENDO HORIZONTALMENTE,"
1040 PRINT "MAIS DANOS VAI CAUSAR."
1050 GOSUB 1110
1060 PRINT "VOCE SERA CAPAZ DE"
1070 PRINT "DERROTAR O BRUXO A TEMPO"
1080 PRINT "DE SALVAR OS MILHARES"
1090 PRINT "DE PRISIONEIROS?"
```

```
 1100 RETURN
 1110 PRINT "APERTE UMA TECLA ";
●1120 IF INKEY$="" THEN GOTO 1120
●1130 CLS
 1140 RETURN
 1150 PRINT "VOCE CONSEGUIU ABRIR"
 1160 PRINT "UM BURACO NO MURO"
 1170 PRINT "DO BRUXO. VOCE"
 1180 PRINT "LIBERTOU TODAS"
 1190 PRINT "AS VITIMAS."
 1200 STOP
```

A listagem acima é para o TK-83. Para os outros computadores, faça as mudanças abaixo.

```
 ■ todos os RND para RND(0)
 ● todos os RND para RND(1)
 ● 10,450,1130 HOME
■●   60 IF LEFT$(I$,1)="S" THEN GOSUB 740
■●  140 LET W(X,Y)=29
■●  430 FOR I=1 TO 200
■●  490 PRINT CHR$(W(X,Y)+32);
 ● 1120 GET I$
```

Como tornar o jogo mais difícil

Você pode aumentar a espessura do muro mudando as seguintes linhas:

Linha 100: troque o primeiro número por um número maior, 10, por exemplo.

Linha 130: subtraia 4 do número que usou na linha 100 e coloque o resultado no final da linha 130.

Linha 480: troque o último número para o mesmo número que usou na linha 100.

Se você tornar o muro muito mais grosso, precisará de mais tempo para perfurá-lo. Assim, é melhor trocar o último número da linha 180 por um número maior.

Míssil!

Este jogo é diferente dos anteriores, pois utiliza instruções gráficas. Como essas instruções variam muito de micro para micro, apresentamos uma listagem separada para cada modelo. Leia a descrição do jogo nesta página e procure nas páginas seguintes a versão apropriada ao seu computador.

Como jogar Míssil!

Você dispõe de três bases, cada uma capaz de lançar um míssil. Ao ver um avião se aproximando, você deve estimar a sua altitude e velocidade e lançar os mísseis contra ele.

Para lançar um míssil, basta apertar uma tecla qualquer. Primeiro é lançado o míssil da esquerda, depois o do centro e depois o da direita.

Veja quantos aviões inimigos você consegue derrubar.

Míssil!: Versão TRS-80

```
 10 CLS
 20 DIM Y(3),F(3)
 30 N=1
 40 PS=INT(RND(0)*3+1)
 50 P=INT(RND(0)*36+5)
 60 GOSUB 400
 70 FOR I=PS TO 100 STEP PS
 80 GOSUB 300
 90 F$=INKEY$
100 IF F$="" OR N>3 THEN 130
110 F(N)=1
120 N=N+1
130 FOR J=1 TO 3
140 RESET(32*J,46-Y(J))
150 IF F(J)=0 OR Y(J)>45 THEN 190
160 Y(J)=Y(J)+1
170 IF POINT(32*J,46-Y(J)) THEN 230
180 SET (32*J,46-Y(J))
190 NEXT
200 NEXT
210 PRINT @0,"ERROU"
220 END
230 PRINT @0,"IMPACTO!!!"
240 END
300 RESET (I-PS,P) : RESET (I-PS+1,P)
310 SET (I,P) : SET (I+1,P)
320 RETURN
400 FOR J=1 TO 3
410 SET (J*32,47)
420 SET (J*32+1,47)
430 NEXT
440 RETURN
```

Míssil!: Versão Apple

```
 10 HOME
 20 HGR
 30 HCOLOR=3
 40 DIM Y(3),F(3)
 50 N=1 : MS=5
 60 PS=INT(RND(1)*6+4)
 70 P=INT(RND(1)*135+11)
 80 GOSUB 400
 90 FOR I=PS TO 265 STEP PS
100 X=I-PS : Y=159-P : C=0 : GOSUB 300
110 X=1 : C=3 : GOSUB 300
120 F$="" : IF PEEK(-16384)>127 THEN GET F$
130 IF F$="" OR N>3 THEN 160
140 F(N)=1
150 N=N+1
160 FOR J=1 TO 3
170 C=0 : GOSUB 350
180 IF F(J)=0 OR Y(J)>145 THEN 230
190 Y(J)=Y(J)+MS
200 C=3 : GOSUB 350
210 X=J*70-I : Y=P-Y(J)
220 IF X>-1 AND X<15 AND Y>-9 AND Y<5 THEN 270
230 NEXT
240 NEXT
250 VTAB 22 : PRINT "ERROU"
260 END
270 VTAB 22 : PRINT "IMPACTO!!!"
280 END
300 HCOLOR=C
310 HPLOT X,Y TO X,Y-8
320 HPLOT TO X+3,Y-2 : HPLOT TO X+12,Y-2
330 HPLOT TO X+14,Y : HPLOT TO X,Y
340 RETURN
350 HCOLOR=C
360 HPLOT 70*J,158-Y(J) TO 70*J,154-Y(J)
370 RETURN
400 FOR J=1 TO 3
410 HPLOT 70*J-5,159 TO 70*J+5,159
420 NEXT
430 RETURN
```

Míssil!: Versão TK-85

```
 10 CLS
 15 INVERSE 0
 20 DIM Y(3)
 30 DIM F(3)
 40 LET N=1
 50 LET MS=8
 60 LET PS=INT(RND*6+5)
 70 LET P=INT(RND*140+20)
 80 GOSUB 400
 90 FOR I=PS TO 240 STEP PS
100 LET C=1 : LET X=I-PS
110 GOSUB 300
120 LET C=0 : LET X=I
130 GOSUB 300
140 LET F$=INKEY$
150 IF F$="" OR N>3 THEN GOTO 170
160 LET F(N)=1 : LET N=N+1
170 FOR J=1 TO 3
180 LET C=1 : GOSUB 350
190 IF F(J)=0 OR Y(J)>148 THEN GOTO 240
200 LET Y(J)=Y(J)+MS
210 LET C=0 : GOSUB 350
220 LET X=J*64-I : LET Y=P-Y(J)
230 IF X>-1 AND X<12 AND Y<10 AND Y>-5 THEN GOTO 280
240 NEXT J
250 NEXT I
260 PRINT AT 0,0;"ERROU"
270 STOP
280 PRINT AT 0,0;"IMPACTO!!!"
290 STOP
300 INVERSE C
310 PLOT X,P
320 DRAW 0,8 : DRAW 3,-6
330 DRAW 8,0 : DRAW 2,-2
340 DRAW -13,0 : RETURN
350 INVERSE C
360 PLOT 64*J+4,Y(J)+4
370 DRAW 0,6 : DRAW -2,-2
380 DRAW 2,2 : DRAW 2,-2
390 RETURN
400 FOR A=65 TO 66
410 FOR B=0 TO 7
420 READ C
430 POKE USR CHR$(A)+B,C
440 NEXT B
450 NEXT A
460 FOR J=1 TO 3
470 PRINT AT 21,8*J;"[AB]"
480 NEXT J
490 RETURN
500 DATA 0,0,0,0,255,255,
    127,63
510 DATA 0,252,252,252,
    255,255,254,252
```

Míssil!: Versão TK-83

```
 10 CLS
 20 DIM Y(3)
 30 DIM F(3)
 40 LET N=1
 50 LET P=INT(RND*19+2)*2
 60 FOR I=1 TO 48
 70 PLOT I,P
 80 PLOT I+1,P
 90 UNPLOT I-1,P
100 LET B$=INKEY$
110 IF B$="" OR N>3 THEN GOTO 140
120 LET F(N)=1
130 LET N=N+1
```

```
140 FOR J=1 TO 3
150 IF F(J)=0 THEN GOTO 190
160 UNPLOT J*16,Y(J)
170 IF Y(J)=42 THEN GOTO 210
180 LET Y(J)=Y(J)+1
190 PLOT J*16,Y(J)
200 IF P=Y(J) AND (J*16=I OR J*16=I+1)
    THEN GOTO 260
210 NEXT J
220 NEXT I
230 CLS
240 PRINT "ERROU"
250 STOP
260 CLS
270 PRINT "IMPACTO"
280 STOP
```

Nota para os usuarios do TK-85:
Os caracteres graficos sao indicados na listagem de acordo com a seguinte convencao: um caractere [significa aperte a tecla GRAPHICS e depois escreva as letras ou numeros seguintes com a tecla SHIFT apertada ate chegar a um]'. Entao aperte de novo a tecla GRAPHICS, voltando ao modo normal.

Algumas sugestões

Aqui estão algumas idéias que você poderá usar nos programas deste livro ou em seus próprios programas.

Ao incluir novas instruções em um programa já pronto, você tem duas opções: utilizar apenas linhas vagas ou numerar de novo todo o programa. No segundo caso, não se esqueça de mudar todas as instruções GOTO e GOSUB.

Faça o computador explicar como se joga

Você pode fazer com que o computador imprima instruções sobre o jogo. Para isso, basta acrescentar algumas linhas no início do programa, como no exemplo abaixo, e colocar uma sub-rotina no final.

```
   10 PRINT "NOME DO JOGO"
   11 PRINT "VOCE QUER SABER"
   12 PRINT "COMO JOGAR?"
   15 INPUT I$
 TK17 IF I$(1)="S" THEN GOSUB 1000
■● 17 IF LEFT$(I$,1)="S" THEN GOSUB 1000
```

O programa principal entra aqui

```
1000 PRINT "VOCE TEM QUE FAZER"
1010 PRINT "O SEGUINTE : ....."
1999 RETURN
```

Você pode usar quantas linhas quiser para as instruções. Não se esqueça de colocar um número e a palavra PRINT no início de cada linha. O número de caracteres entre aspas deve ser menor ou igual ao número de caracteres que o seu computador pode imprimir em uma linha. Não se esqueça também de colocar uma linha RETURN no final da sub-rotina.

Faça o computador parar e esperar por você

Se houver muitas linhas de instruções, talvez você precise incluir a sub-rotina abaixo, que faz com que o computador pare de executar o programa até você apertar uma tecla. Assim, evitará que as instruções desapareçam na parte superior da tela antes que tenha tempo de lê-las. Coloque uma linha GOSUB no ponto em que deseja que o programa pare e acrescente no final a seguinte sub-rotina:

```
   1000 PRINT "APERTE UMA TECLA"
   1010 PRINT "PARA CONTINUAR"
■TK1020 IF INKEY$="" THEN GOTO 1020
 ●1020 GET I$
   1030 PRINT
   1040 RETURN
```

Faça o computador "conversar" com você

Você pode programar o computador para fazer perguntas e agir de acordo com as suas respostas. Aqui está, por exemplo, um pequeno programa que faz com que o computador se recuse a jogar, a menos que seu nome comece com a letra J:

```
    1 PRINT "COMO VOCE SE CHAMA?"
    2 INPUT I$
  TK3 IF I$(1)<>"J" THEN GOTO 1000
■● 3 IF LEFT$(I$,1)<>"J" THEN GOTO 1000
    4 PRINT "OK... VOCE PODE JOGAR"
    5 PRINT "ESTA PRONTO?"
    6 INPUT J$
  TK7 IF J$(1)<>"S" THEN GOTO 5
■● 7 IF LEFT$(J$,1)<>"S" THEN GOTO 5
```

O programa principal entra aqui

```
1000 PRINT "DESCULPE.ESTE JOGO E"
1010 PRINT "SO PARA PESSOAS"
1020 PRINT "CUJOS NOMES COMECAM"
1030 PRINT "COM A LETRA J"
```

Aqui está um programa em que o computador desconfia de que você não tem coragem suficiente para jogar.

```
  10 PRINT "JOGO MUITO PERIGOSO"
  12 PRINT "VOCE TEM CORAGEM"
  14 PRINT "DE ENFRENTAR"
  15 PRINT "O MONSTRO CABELUDO?"
  16 INPUT I$
TK17 IF I$(1)="S" THEN GOTO 20
■●17 IF LEFT$(I$,1)="S" THEN GOTO 20
  18 PRINT "COVARDE"
  19 STOP
```

Você pode combinar o programa acima com a sub-rotina de instruções, colocando as linhas 11 a 17 do programa da página anterior como linhas 20 a 26 do programa acima. Comece o programa principal na linha 30 e acrescente a sub-rotina de instruções no final.

Quer jogar de novo?

Em lugar de escrever RUN cada vez que deseja começar um jogo, você pode fazer com que o computador pergunte se quer jogar de novo. Coloque as linhas abaixo no final do programa, antes do último STOP.

```
  1000 PRINT "QUER JOGAR DE NOVO?"
  1010 INPUT I$
TK1020 IF I$(1)="S" THEN RUN
  1020 IF LEFT$(I$,1)="S" THEN RUN
  1030 PRINT "OK...TCHAU"
```

Mude os números das linhas de acordo com o seu programa.

Efeitos sonoros

Muitos microcomputadores dispõem de recursos para gerar sons. Você pode utilizar esses recursos em qualquer ponto dos seus programas. Experimente, por exemplo, simular o ruído de uma explosão ou fazer com que o computador toque uma musiquinha toda vez que alguém ganhar. Como as instruções para gerar sons variam de computador para computador, você terá que consultar o manual do seu micro. Em certos casos, você pode acrescentar uma única linha ao programa no local onde deseja o som. Em outros, vai precisar de várias linhas; nesse caso, é melhor usar uma sub-rotina.

A linha abaixo mostra, por exemplo, a instrução para produzir o som do dó central do piano durante 6 segundos em um computador do tipo TRS-80:

```
SOUND 89,100
```

Você pode usar quantas instruções quiser para gerar sons, mas não se esqueça de colocar um número no início de cada instrução.

Nota especial para os usuários do TK-85

Se o seu micro é um TK-85, talvez alguns dos jogos deste livro sejam rápidos demais para o seu computador. As instruções para diminuir a velocidade aparecem na mesma página que o jogo correspondente. Como regra geral, quanto maior o número que você usar no loop de espera, mais lento será o jogo.

Como programar jogos

Depois que você estiver familiarizado com os jogos deste livro, certamente terá vontade de programar outros jogos. Nestas duas páginas, discutimos alguns conceitos básicos de que vai necessitar para isso.

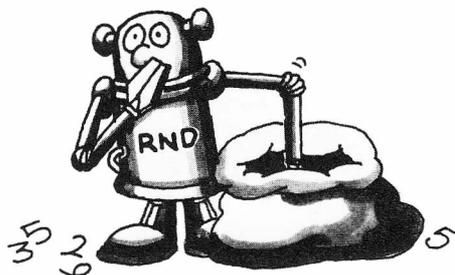
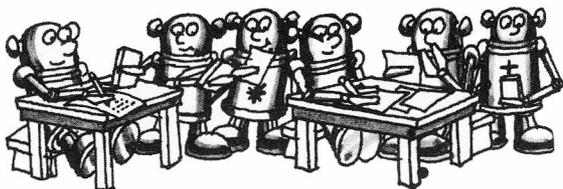
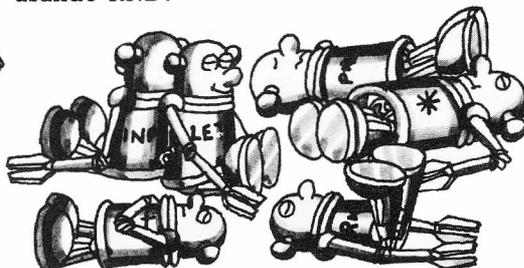
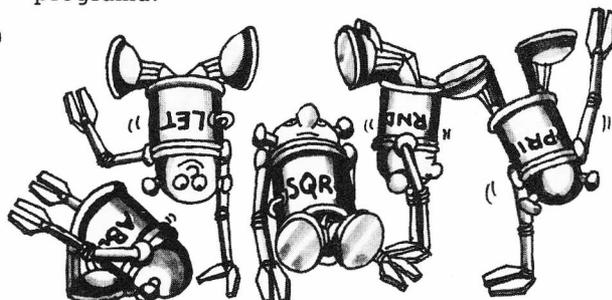
Antes de mais nada, vale a pena pensar um pouco no que o seu computador pode fazer e no que ele não pode.

*Pode guardar informações.

*Pode fazer contas.

*Pode tomar decisões comparando de várias formas informações anteriores.

*Pode mostrar os resultados dos cálculos e decisões e também o que está guardado na memória.

*Pode pedir novas informações.

*Pode escolher um número ao acaso, usando RND.

*Não pode fazer nada que não esteja no programa.

*Se você usar instruções válidas de BASIC, não pode distinguir instruções "sensatas" de instruções tolas.

Quando estiver planejando o programa de um jogo, não inclua nada que o seu micro não seja capaz de fazer.

Como planejar um jogo

Antes de programar o computador para jogar um jogo, você precisa saber exatamente como será jogado e quais serão as regras. Como o computador vai necessitar de uma série de instruções simples e lógicas, é melhor imaginar primeiro todo o jogo e depois dividi-lo em pequenas partes.

Em seguida, você deve escrever todas as partes do jogo, na ordem em que serão executadas (escreva em português; ainda não está na hora de usar BASIC).

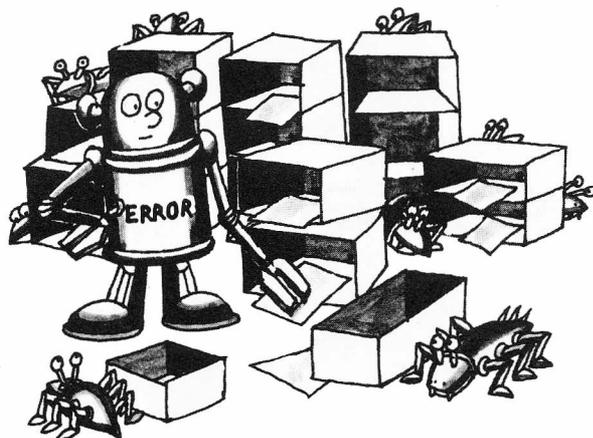
A ilustração abaixo mostra o plano de um jogo simples de atirar, que pode ser usado para disparar balas de canhão contra um navio pirata ou raios laser contra uma nave alienígena.

PLANO DE JOGO

1) IMPRIMIR TÍTULO E INSTRUÇÕES
2) ESCOLHER UM ALVO
3) COMEÇAR UM LOOP QUE PERMITA AO JOGADOR N TENTATIVAS
4) PEDIR AO JOGADOR PARA DAR UM TIRO
5) VERIFICAR SE O TIRO ACERTOU
6) IMPRIMIR UMA MENSAGEM SOBRE O RESULTADO DO TIRO
7) VOLTAR PARA A TENTATIVA SEGUINTE

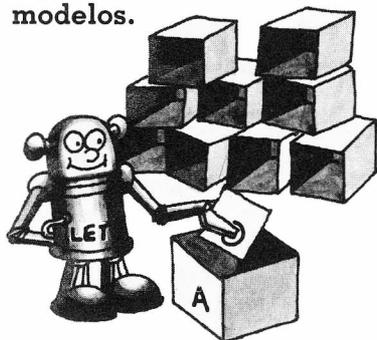
Como escrever o programa

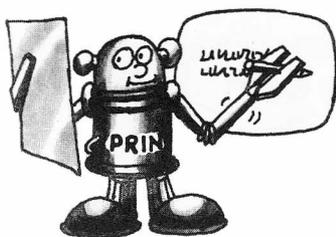
O passo seguinte consiste em traduzir o seu plano para a linguagem BASIC. Cada parte do plano pode corresponder a várias linhas de BASIC. Deixe números vagos entre os números das linhas do programa, para poder acrescentar mais tarde novas instruções, se for necessário.

Escreva um rascunho do programa e comece a testá-lo no computador. O computador é capaz de detectar muitos erros de programação e mesmo, em certos casos, de indicar exatamente onde está o erro. Se o seu programa não der certo logo da primeira vez, não desanime; "acertar" um programa é um processo lento e tedioso, até mesmo para os programadores experientes.

Depois que o programa básico estiver funcionando, você pode aperfeiçoá-lo, acrescentando mensagens, comentários, alvos diferentes etc. Também pode incluir partes dos programas deste livro nos seus jogos.

Não espere que os seus primeiros jogos sejam muito movimentados e originais. Comece com idéias simples e esteja preparado para modificá-las se for necessário. É provável que você descubra que incluiu alguma coisa no jogo que é muito fácil para uma pessoa mas muito difícil para um computador. Quando adquirir experiência, saberá instintivamente o que o seu computador é capaz de fazer.

Instruções de BASIC

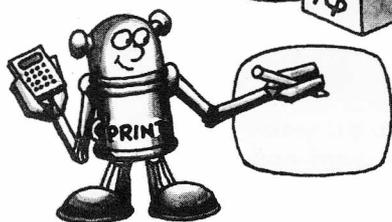
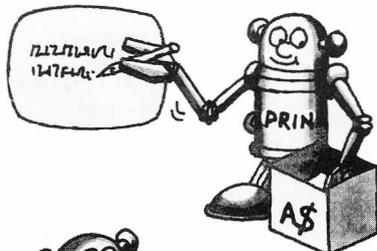
Este capítulo explica como funcionam algumas instruções comuns de BASIC. Quase todas foram usadas nos programas deste livro, de modo que você pode consultar as listagens dos jogos para ver o que essas instruções fazem nos programas. Nem todas as instruções podem ser usadas em todos os micros mencionados neste livro. A tabela de equivalência da página 44 mostra as diferenças entre os vários modelos.

LET diz ao computador para dar um nome a uma parte da memória e guardar nesta parte um número que você especifica. Assim, por exemplo, LET A = 6 significa que o computador deve chamar de "A" uma parte da memória e guardar nesta parte o número 6. "A" é chamado de "variável" e colocar alguma coisa em A é chamado de "atribuir um valor a uma variável".

Alguns nomes de variáveis são seguidos por um cifrão (A$, por exemplo). Isto significa que a variável é do tipo string e a memória correspondente pode conter um número qualquer de caracteres (letras, números e símbolos).

PRINT diz ao computador para mostrar alguma coisa na tela. Esta instrução pode ser usada de várias formas:

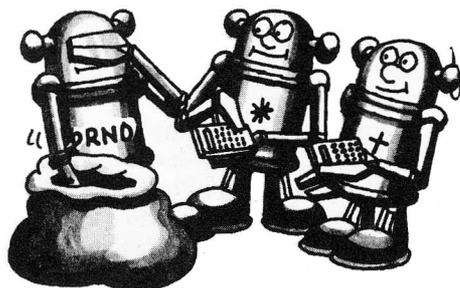
Uma mensagem entre aspas precedida por PRINT é mostrada na tela exatamente como foi escrita. A parte entre aspas não precisa ser em BASIC, pode ser qualquer coisa.

A instrução PRINT seguida pelo nome de uma variável (PRINT A ou PRINT A$, por exemplo) diz ao computador para mostrar na tela o valor da variável.

PRINT também pode ser usada para fazer cálculos e mostrar o resultado. Assim, por exemplo, a instrução PRINT 6*4 faz o computador mostrar na tela o número 24.

Você pode usar PRINT sozinha para deixar uma linha em branco na tela.

RND diz ao computador para escolher um número ao acaso. Computadores diferentes usam diferentes formas de RND (essas diferentes formas estão na tabela da pág. 44). Nos computadores TK, RND gera um número entre 0 e 0,99999999. Você pode mudar os limites dos números gerados por RND multiplicando-os por uma constante e somando-os a outra constante. Assim, por exemplo, RND*20 gera um número entre 0 e 19,99999999, enquanto que RND*20 + 1 gera um número entre 1 e 20,99999999.

Para gerar apenas números inteiros, use a instrução INT.

Para gerar letras e símbolos, use a instrução CHR$.

INT é a abreviação de inteiro. No caso de números positivos, diz ao computador para ignorar tudo que está à direita do ponto decimal. Assim, por exemplo, INT (20.999) é o número 20. No caso de números negativos, tudo que está à direita do ponto é ignorado e o número que está à esquerda é "aumentado" de uma unidade. Assim, por exemplo, INT(-3.6) é o número -4.

A instrução INT é freqüentemente usada junto com RND, como em INT(RND*20+1), que diz ao computador para escolher ao acaso um número inteiro entre 1 e 20.

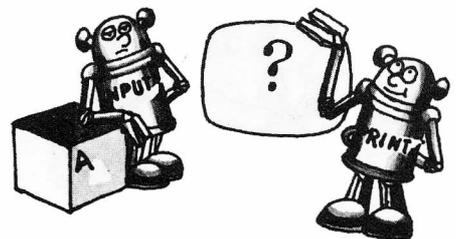
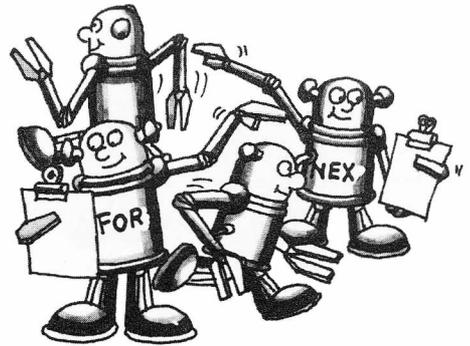
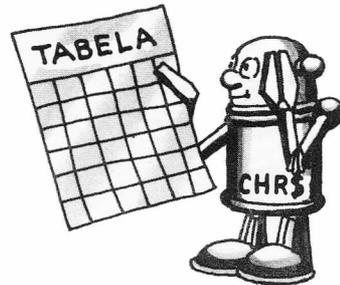
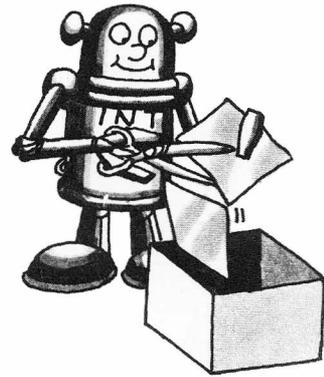
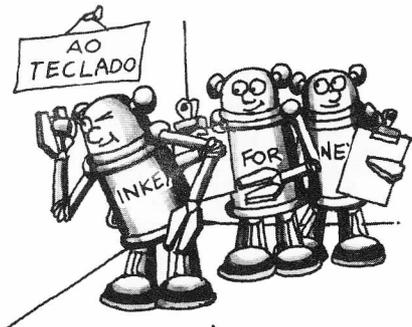
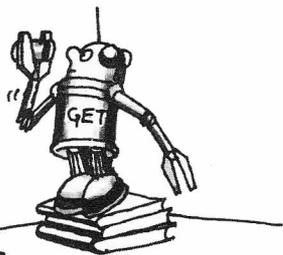
CHR$ é usada para transformar números em letras. Com exceção do TK-83, todos os computadores mencionados neste livro utilizam o código ASCII* para a correspondência entre números e caracteres. Assim, por exemplo, o número de código para a letra A é 65 e a instrução CHR$(65) faz a letra A aparecer na tela.

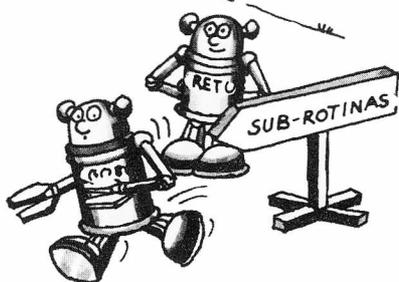
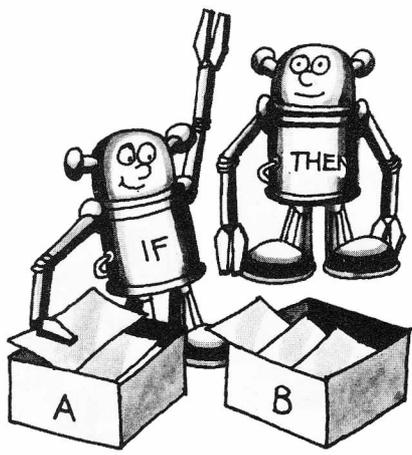
Você pode usar as instruções CHR$, INT e RND para fazer o computador escolher letras ao acaso, como em CHR$(INT(RND*26+65)), que no computador TK-85 gera uma letra qualquer entre A e Z (para os outros computadores, consulte a tabela da página 44).

FOR é usada para começar um loop que faz o computador repetir uma parte do programa um certo número de vezes. Deve ser seguida pelo nome de uma variável (G, por exemplo) e pelos valores inicial e final da mesma variável (= 1 TO 10, por exemplo).

O final do loop é indicado pela instrução NEXT (NEXT G, neste exemplo), que faz o valor da variável aumentar de uma unidade e manda o computador de volta para a instrução FOR. Quando o valor da variável chega ao valor final, o computador ignora a instrução NEXT e passa para a linha seguinte. Para cada instrução FOR deve haver uma instrução NEXT, caso contrário o programa não funcionará.

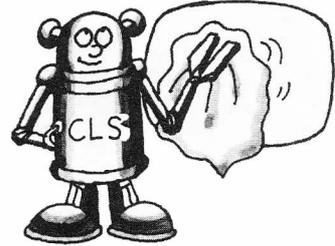
INPUT diz ao computador para dar um nome a uma variável, mostrar na tela um ponto de interrogação e esperar que você escreva o valor da variável. A execução do programa só prossegue se você apertar a tecla RETURN, ENTER ou NEWLINE.

Você pode usar variáveis numéricas (sem o cifrão) ou variáveis tipo string (com um cifrão no final) em uma instrução INPUT, mas se usar uma variável numérica o computador não aceitará letras.

*O código completo está na pág. 43.

**INKEY$** diz ao computador para verificar se você está apertando alguma tecla e, em caso afirmativo, qual é essa tecla. O computador não espera você apertar alguma tecla, como no caso da instrução INPUT. A instrução INKEY$ é geralmente usada dentro de um loop que faz o computador verificar muitas vezes o teclado. Isto é necessário porque os computadores funcionam muito depressa e você não teria tempo de apertar uma tecla no tempo que o computador leva para fazer uma única verificação.

Quando o loop termina sem que você tenha apertado nenhuma tecla, o computador prossegue com uma string que não contém nenhum símbolo (a chamada string vazia).

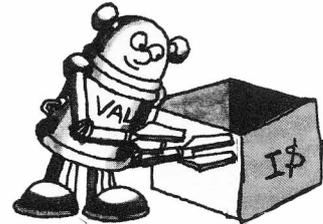
GOTO faz o computador pular para uma instrução mais acima ou mais abaixo, ignorando as instruções intermediárias. GOTO deve sempre ser seguida pelo número da linha para onde o computador deve pular.

GOSUB diz ao computador para sair do programa principal e ir para uma sub-rotina. GOSUB deve ser seguida pelo número da primeira linha da sub-rotina. A última instrução de qualquer sub-rotina deve ser RETURN, que manda o computador de volta ao programa principal, para a linha seguinte à da instrução GOSUB.

IF...THEN diz ao computador para verificar se uma expressão é verdadeira ou falsa e fazer coisas diferentes, dependendo da resposta. É usada com os seguintes sinais, além das palavras AND e OR:

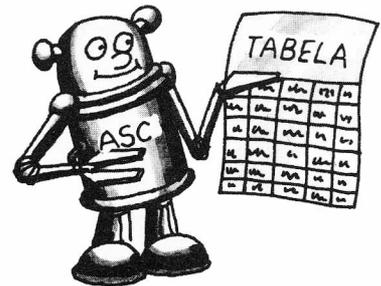
= igual a
< menor que
> maior que
<= menor ou igual a
>= maior ou igual a
<> diferente de

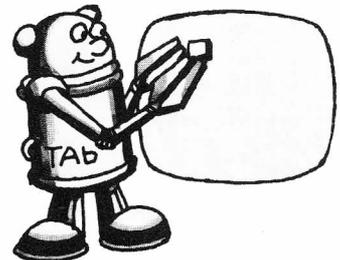
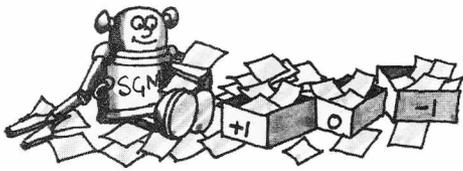
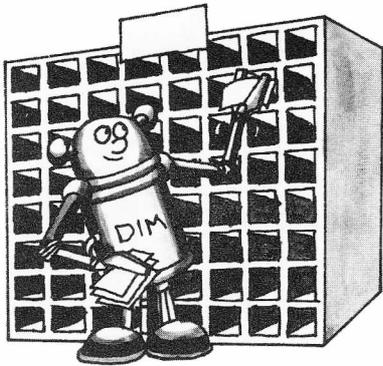
Se o computador verifica que a expressão é verdadeira, executa a instrução que se segue à palavra THEN. Se a expressão é falsa, ignora o resto da linha e passa para a linha seguinte.

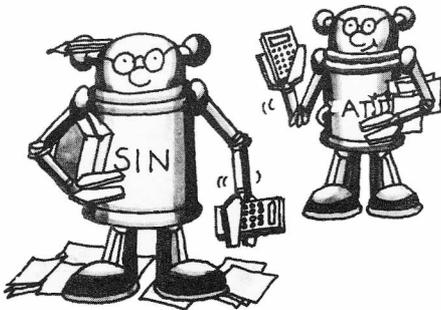
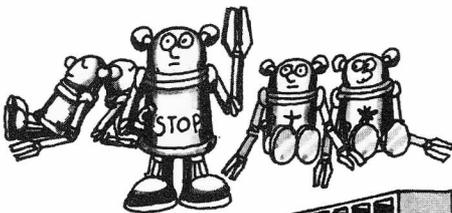
CLS é usada para apagar tudo que está na tela, sem afetar o que se encontra na memória do computador. É útil para limpar a tela antes de executar um programa ou para manter uma mensagem na tela apenas por um período limitado de tempo durante um jogo.

HOME é usada para limpar a tela nos computadores do tipo Apple, em lugar de CLS.

ABS diz ao computador para ignorar os sinais de mais e de menos na frente dos números e tomar o seu valor "absoluto". Assim, por exemplo, ABS(-10) é 10 e ABS (+ 10) também é 10.

VAL é usada para obter o valor numérico de um número associado a uma variável string. Em outras palavras, a instrução VAL diz ao computador para ignorar o cifrão e considerar a variável como se fosse uma variável numérica. Assim, por exemplo, se I$ = "60", então VAL(I$) é o número 60.

ASC é usada para obter o número do código ASCII correspondente a um dado número, letra ou símbolo. Assim, por exemplo, ASC ("3") é igual a 51. A expressão entre parênteses deve ser do tipo string, como "20" ou **A$**.

Os micros da família Sinclair não têm a instrução ASC, embora o TK-85 use o código ASCII.

CODE é a instrução usada nos micros da família Sinclair em lugar de ASC. Como no caso da instrução ASC, CODE deve ser sempre seguida por uma expressão do tipo string, entre parênteses. Observe que os números de código dos caracteres no micro TK-83 não são os mesmos que nos outros computadores.

TAB é uma instrução usada para mover o cursor na tela até a coluna desejada. Geralmente é usada em conjunto com PRINT para mostrar alguma coisa no meio da tela. O número de espaços de deslocamento do cursor é colocado entre parênteses depois de TAB. O número máximo que pode ser usado depende do modelo de computador (vide tabela na pág. 43).

SGN diz ao computador para verificar o sinal de um número. O resultado é -1 se o número é negativo, 0 se o número é zero e +1 se o número é positivo. Assim, por exemplo, SGN (-30) é -1, SGN(7) é +1 e SGN(0) é 0.

DIM diz ao computador para reservar espaço na memória para uma matriz (variável indexada). Assim, por exemplo, DIM X(6) diz ao computador para reservar um espaço suficientemente grande para conter 6 elementos, que serão chamados de X(1), X(2), X(3), etc. DIM A(8,8) reserva um espaço na memória para 64 elementos, que serão chamados de A(1,1), A(1,2), etc. Uma variável indexada nunca pode ser usada em um programa com um número entre parênteses maior que o especificado na instrução DIM.

SQR é usada para obter a raiz quadrada de um número. Assim, por exemplo, SQR(16) é igual a 4.

SIN é usada para calcular o seno de um ângulo. Em um triângulo retângulo, o comprimento do lado oposto a um ângulo dividido pelo comprimento da hipotenusa (lado oposto ao ângulo reto) é chamado de seno desse ângulo. Para usar SIN em um programa, o ângulo deve ser expresso em radianos, e não em graus.

ATN é uma das funções trigonométricas que os computadores são capazes de calcular (a outra é o seno, de que falamos acima). ATN é a abreviação de arco tangente. Em alguns computadores, esta instrução é chamada de ARCTAN.

STOP diz ao computador para parar de executar o programa. Em todos os computadores, a não ser o TK-83, a instrução END faz o mesmo efeito.

PEEK é usada para descobrir o que existe em uma certa região da memória do computador. Esta instrução é sempre seguida por um número entre parênteses, que especifica um "endereço" na memória.

POKE é uma forma especial de guardar informações na memória do computador, usando o "endereço" de uma memória.

Código ASCII

Número de código	Caractere correspondente	Número de código	Caractere correspondente
32	espaço	62	>
33	!	63	?
34	"	64	@
35	#	65	A
36	$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	*	72	H
43	+	73	I
44	,	74	J
45	–	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		

Código TK-83

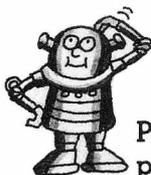
Número de código	Caractere correspondente	Número de código	Caractere correspondente
11	"	41	D
12	£	42	E
13	$	43	F
14	:	44	G
15	?	45	H
16	(	46	I
17	)	47	J
18	>	48	K
19	<	49	L
20	=	50	M
21	+	51	N
22	–	52	O
23	*	53	P
24	/	54	Q
25	;	55	R
26	,	56	S
27	.	57	T
28	0	58	U
29	1	59	V
30	2	60	W
31	3	61	X
32	4	62	Y
33	5	63	Z
34	6		
35	7		
36	8		
37	9		
38	A		
39	B		
40	C		

Tabela de tamanhos de tela

	Número máximo de caracteres na horizontal (ou número de colunas)	Número máximo de caracteres na vertical (ou número de linhas)
TRS-80	64	16
TK-83	32	22
TK-85	32	22
Apple	40	25

Tabela de equivalência

Esta tabela mostra algumas das variações nas instruções de BASIC usadas pelos micros mencionados neste livro. Não foram incluídas as instruções para gerar gráficos, sons e cores, pois variam de fabricante para fabricante. Observe também que embora quase todos os micros usem as instruções PEEK e POKE, os endereços da memória variam de acordo com o fabricante, de modo que os números usados com essas instruções variam de computador para computador.

	Apple	TRS-80	TK-85	TK-83
Escolha um número ao acaso entre 0 e 0,99999999	RND(1)	RND(0)	RND	RND
Escolha um número ao acaso entre 1 e N	RND(1)*N+1	RND(N)	RND*N+1	RND*N+1
Escolha uma letra ao acaso entre A e Z	CHR$(INT(RND(1)*26+65))	CHR$(RND(26)+ 64)	CHR$(INT(RND*26+65))	CHR$(INT(RND*26+38))
Limpe a tela	HOME	CLS	CLS	CLS
Verifique a tecla que está sendo apertada	X$=" " IF PEEK(-16384) >127 THEN GET X$	INKEY$	INKEY$	INKEY$
Transforme caracteres em números de código	ASC("X") (código ASCII)	ASC("X") (código ASCII)	CODE ("X") (código ASCII)	CODE ("X") (código TK-83)
Mova o cursor para cima	CALL – 993	PRINT CHR$(27)	PRINT CHR$(11)	PRINT CHR$(112)
Mova o cursor para baixo	PRINT CHR$(10)	PRINT CHR$(26)	PRINT CHR$(10)	PRINT CHR$(113)
Mova o cursor para a esquerda	PRINT CHR$(8)	PRINT CHR$(24)	PRINT CHR$(8)	PRINT CHR$(114)
Mova o cursor para a direita	PRINT CHR$(21)	PRINT CHR$(25)	PRINT CHR$(9)	PRINT CHR$(115)
Separe os primeiros N caracteres de uma string	LEFT$(A$,N)	LEFT$(A$,N)	A$(1 TO N)	A$(1 TO N)
Separe os últimos N caracteres de uma string	RIGHT$(A$,N)	RIGHT$(A$,N)	A$(N TO)	A$(N TO)
Separe N caracteres do meio de uma string	MID$(A$,N1,N2)	MID$(A$,N1,N2)	A$(N1 TO N2)	A$(N1 TO N2)

Respostas

Pode ser que as suas respostas para alguns problemas sejam diferentes das que aparecem aqui. Isto não importa, contanto que funcionem no seu computador. Entretanto, verifique se são tão simples e concisas como as respostas do livro.

Página 5
Bomba-Relógio

A linha 90 diz ao computador quantas vezes deve repetir o loop e pedir um palpite. Assim, para aumentar o número de tentativas, troque o último número da linha 90 por um número maior. Para diminuir o número de tentativas, use um número menor.

Página 7
Mensagem Decisiva

Neste programa, as linhas 150 e 160 constituem um loop de espera. Elas fazem o computador ficar "parado" um certo tempo antes de passar para a instrução seguinte, que é limpar a tela. Para fazer com que a mensagem fique mais tempo na tela, você precisa aumentar o número de vezes que o loop é repetido. Para isso, basta trocar o último número da linha 150 por um número maior.

Página 9
Duelo

Para que o computador possa também errar o tiro, acrescente as linhas abaixo, além das que aparecem em "Como tornar o jogo mais difícil".

```
155 IF RND<.1 THEN GOTO 250
250 PRINT "ELE ATIROU, MAS ERROU"
260 GOTO 90
```

A linha 155 pode ter RND(1) ou RND(0), dependendo do seu computador.

Página 11
Batalha no Deserto

Para incluir a possibilidade de os robôs atirarem em você, acrescente as linhas abaixo.

```
175 IF RND<.05 THEN GOTO 250
250 PRINT "ELES VIRAM VOCE...AHHHHHHH!"
260 STOP
```

Use a forma de RND na linha 175 que seja compatível com o seu computador. Você pode substituir .05 por qualquer número entre 0 e 0.999, mas não se esqueça de que quanto maior for o numero, maior a probabilidade de os robôs atirarem em você.

Página 13
O Castelo do Traidor

Este é um problema muito difícil; não se aborreça se não tiver conseguido resolvê-lo sozinho. De qualquer forma, vale a pena experimentar a solução no programa. Acrescente as linhas abaixo para fazer com que apareçam dois tipos de alvos.

```
 55 LET P=INT(RND+.5)*4+1
 57 LET P$=CHR$(P+51)
▲■● 57 LET P$=CHR$(P+78)
 70 IF L=T THEN LET R$=R$+P$
100 LET S=S+P
```

Página 15
Invasão dos Robôs

Para ganhar 100 pontos pelos robôs U, V, W, X e Y e 10 pontos pelos outros, faça as modificações abaixo.

```
220 LET H=H+10
225 IF P$>"T" AND P$<"Z" THEN LET
    H=H+90
```

Página 17
Arma Secreta

Para acrescentar o sistema de contagem, basta colocar as linhas abaixo no programa.

```
 15 LET S=0
145 IF Z<=3 THEN LET S=S+1
190 GOTO 217
215 LET S=S+10
217 PRINT "VOCE FEZ ";S; " PONTOS"
```

Página 19
Fuga!

Esta é a forma mais simples de incluir três robôs, embora torne o jogo bastante difícil.

```
 22 FOR R=1 TO 3
 25 PRINT "ROBO ";R
300 NEXT R
310 STOP
```

Página 21
Avião Pirata

Para fazer com que você e o pirata tenham a mesma probabilidade de vencer, troque o .7 da linha 290 para .5.

Página 23
Bombardeiro Supersônico

Para aumentar o número de alvos, troque o 5 das linhas 30, 70 e 120 por um número maior. Você deve usar o mesmo número nas três linhas.

Página 25
"Iceberg"

Você pode aumentar o tamanho do mapa trocando o 8 das linhas 20, 50, 70, 80, 110, 120, 160, 170, 290 e 310 por um número maior. (Use o mesmo número em todas essas linhas.)

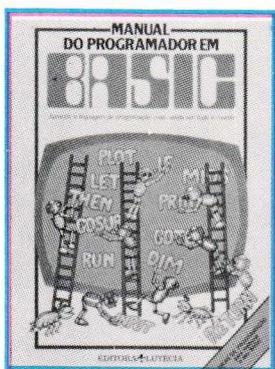
Para aumentar o número de *icebergs*, troque o 4 na linha 30 por um número maior.

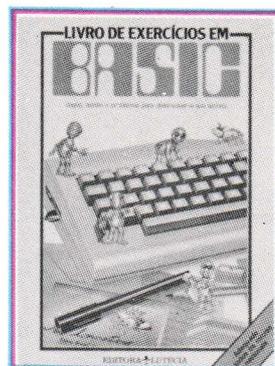
Guias Práticos de Microcomputadores

O que é capaz de fazer um microcomputador! Calcular, evidentemente, mas também organizar perguntas, escrever poemas, jogar uma quantidade de jogos cada qual mais palpitante, até mesmo compor música... Pequenos guias práticos de introdução à microinformática, as obras desta nova coleção nos permitem descobrir todas as possibilidades que os microcomputadores nos oferecem. Eles nos iniciam na linguagem e no funcionamento do computador, na aprendizagem da programação e — por que não? — na criação de programas originais: a clareza dos textos, a alegria das cores, a graça dos desenhos, tudo é concebido nestes livros para fazer desta iniciação um prazer.

Um colorido guia para melhor compreendermos os microcomputadores — como trabalham e o que fazem, apresentando idéias de coisas que podem ser feitas com o micro. Apresenta um utilíssimo guia comparativo dos diversos micros existentes no mercado brasileiro.

Um guia passo a passo de programação para os absolutamente iniciantes. Apresenta as diferentes linguagens e aprofunda no BASIC, incluindo pequenos programas, simples e divertidos. Profusamente ilustrado a cores.

13 programas fascinantes e ao mesmo tempo simples, para todo tipo de microcomputadores; Jogos Intergaláticos, Módulo Lunar, Viagem ao Futuro, Salvamento no Espaço e muitos outros, com respostas dos problemas.

Introduz o leitor na programação BASIC, através de exemplos didáticos, e é sobretudo indicado ao jovem que começa a programar. Substitui com vantagens os manuais dos fabricantes. Apresenta dezenas de dicas utilíssimas para aprimorar a técnica de programação.

13 listagens de programas de jogos, para serem "rodados" nos micros pessoais mais difundidos no Brasil. Muito útil para ensinar o leitor a desenvolver seus próprios programas. Além disso, é um passatempo fascinante.

Um verdadeiro caderno a cores de problemas e exercícios, com jogos e quebra-cabeças, ideais para todo curso de programação em BASIC. Todas as respostas e explicações estão incluídas, para ajudar a esclarecer todo e qualquer problema.

PROGRAMAS DE JOGOS DE GUERRA

LUTÉCIA 92326/8